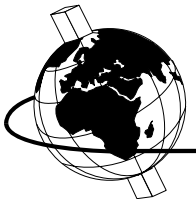


SWIFTBROADBAND SAFETY SERVICES TEST TOOL USER'S GUIDE

DC-350301 V1.1

REVISION	1.0	1.1			
DATE	2012/11/12	2013/04/08			

This document is supplied pursuant to Amendment #1 of contract INM/10-4906/JB between Inmarsat Global Ltd. and Square Peg Communications Inc., dated December 12, 2011. This document contains confidential information which may be used only in the manner specified in that Contract, unless such information is or lawfully comes into the public domain or is or lawfully becomes available to the user on other terms.

Square Peg Communications Inc.
4017 Carling Avenue, Suite 200
Ottawa, Ontario, Canada K2K 2A3
Phone: (613) 271-0044 Fax: (613) 271-3007

TABLE OF CONTENTS

1. INTRODUCTION 1-1
1.1 BACKGROUND 1-1
1.2 SCOPE 1-1
1.3 CONVENTIONS 1-2
1.4 APPLICABLE DOCUMENTS 1-2
1.5 ACRONYMS 1-3
2. ARCHITECTURE 2-1
2.1 TEST CONTROLLER 2-1
2.1.1 Overview 2-1
2.1.2 Platform 2-1
2.1.3 Script Language 2-2
2.2 PHASE 1 TEST ARCHITECTURE 2-3
2.3 PHASE 2 TEST ARCHITECTURE 2-4
2.4 PHASE 3 TEST ARCHITECTURE 2-5
3. INSTALLATION 3-1
3.1 SYSTEM REQUIREMENTS 3-1
3.1.1 D: Drive 3-1
3.2 HARDWARE INSTALLATION 3-2
3.2.1 Installation Of CEI-520A Condor ARINC 429 Card 3-2
3.2.2 LAN Interface 3-2
3.3 SOFTWARE AND SCRIPT INSTALLATION 3-2
3.3.1 Test Controller Software 3-2
3.3.2 Test Scripts 3-3
3.4 INITIALIZATION FILES 3-3
3.4.1 Test Controller Initialization Files 3-4
3.4.2 Test Script Initialization Files 3-4
4. TEST SCRIPT COVERAGE AND ORGANIZATION 4-1
4.1 TEST COVERAGE 4-1
4.2 DIRECTORY STRUCTURE 4-3
5. TEST SCRIPT INTERFACES AND CONFIGURATION 5-1
5.1 INTERFACES 5-1
5.1.1 IP Interface Overview 5-1
5.1.2 Control Interface 5-3
5.1.3 Williamsburg Interface 5-3
5.1.4 IRS Interface 5-3
5.1.5 ARINC Status Interface 5-3
5.1.6 Modem Interface 5-3
5.2 CONFIGURATION PARAMETERS 5-4
6. PHASE 1 OPERATION 6-1
6.1 SETUP 6-1
6.2 SCRIPT AND SUITE FILES 6-2
6.3 INITIALIZATION AND AES LOGON STATUS 6-2

6.4 TEST SCRIPT EXECUTION 6-3
6.5 TEST SCRIPT OUTPUT 6-5
6.6 FAULT ISOLATION 6-6
6.6.1 Connectivity Issues 6-6
6.7 TEST SUITE EXECUTION 6-7
7. PHASE 2 OPERATION 7-1
7.1 SETUP 7-1
7.2 TEST EXECUTION 7-2
7.3 FAULT ISOLATION 7-2
8. PHASE 3 OPERATION 8-1
8.1 SETUP 8-1
8.2 TEST EXECUTION 8-5
8.3 FAULT ISOLATION 8-5

FIGURES

Figure 2-1 Phase 1 Test Configuration 2-3
Figure 2-2 Phase 2 Test Configuration 2-4
Figure 2-3 Phase 3 Test Configuration 2-5
Figure 6-1 Example Phase 1 Test Configuration 6-1
Figure 6-2 Example Run-Time Error Message For Bad IP Address 6-6
Figure 6-3 Example Run-Time Error Message For Overlapping Ports 6-6
Figure 7-1 Example Phase 2 Test Configuration 7-1
Figure 8-1 Example Phase 3 Test Configuration 8-1

TABLES

Table 2-1 Test Controller Functions 2-1
Table 3-1 BSCR Software Update 3-2
Table 3-2 SBB Script Installation 3-3
Table 4-1 Testcase Summary For Phase 1/2/3 Configuration 4-1
Table 4-2 Directory Structure 4-3
Table 5-1 UDP Ports 5-1
Table 5-2 Test Script Configuration Parameters 5-4
Table 6-1 Typical Set of Parameters to be Defined in D:\Bplt_Dat\SBB\Ini\SBB_Par.ini for Phase 1 6-1
Table 7-1 Typical Set of Parameters to be Defined in D:\Bplt_Dat\SBB\Ini\SBB_Par.ini for Phase 2 7-2
Table 8-1 Typical Set of Parameters to be Defined in D:\Bplt_Dat\SBB\Ini\SBB_Par.ini for Phase 3 8-2

REVISIONS

Version	Date	By	Revision
1.0	2012/11/12	N. Robin M. Gertsman	Initial draft for external review.
1.1	2013/01/23	M. Gertsman	Table 5-1: Update AIGI source/destination ports. Section 5.1.1: Clarify use of primary AGGW IP address and restrictions on port assignments. Table 5-2: Remove [SBB005_103]TestNoGwConfOnly. Table 5-2: Add [Arinc429]IrsDelayMs, [Arinc429]DisconnectWbpAtStartOfEachTest [Sdu]LocationSource [SBB002_113]Enable = TRUE Table 5-2: Replace [Aigi]AigiSrcPort and AigiDestPort with [Aigi]AagwSrcPort, AagwDestPort, MinAggwSrcPort, MaxAggwSrcPort, AggwDestPort. Section 6.6: Add more information regarding isolating connectivity issues.
	2013/01/29	M. Gertsman	Table 5-2: Clarify [TestControl]RestartAagwEachTest.
	2013/01/30	M. Gertsman	Section 5.1.1: Update use of primary AAGW address and ARINC 429 local IP address. Table 5-2: Add [Arinc429]LocalIpAddress.
	2013/04/08	M. Gertsman	Table 3-2, Table 4-2: Add Doc directory. Table 5-2: Update default [Aigi]MinAggwSrcPort. Section 8: Minor tweaks to text. Release.

1. INTRODUCTION

This document provides instructions for the installation, configuration and use of the Inmarsat SBB Safety Test Tool that exercises and validates the ACARS Aircraft Gateway (AAGW) interface to the ACARS Ground Gateway (AGGW) over the SwiftBroadband network.

1.1 Background

Inmarsat plans to offer aeronautical safety services via its SwiftBroadband system. In concert with its manufacturing partners, Inmarsat has developed an annex to the BGAN System Definition Manual (SDM) which specifies the safety service related additions to the SwiftBroadband (SBB) network required for supporting SwiftBroadband Oceanic Safety Services.

The additional services are:

- ACARS
- Voice
- Prioritised IP

The SDM specifies an ACARS Aircraft Gateway (AAGW) which manages the terminal's ACARS connectivity with the ground via either SBB or Classic Aero. Terminal manufacturers implementing support for SBB Safety Services must verify that their AAGW implementations comply with Inmarsat's requirements as specified in the SDM [1].

Inmarsat have indicated that the initial Type Approval test strategy will be based on a two stage approach:

1. Test the ACARS Intra-Gateway Interface (AIGI) with error injection (protocol verification), and verify AIGI message format and content along with the transport of ACARS messages, using an IP link;
2. Test the AIGI over the air against the real AGGW (without the use of the test tool).

Inmarsat contracted Square Peg Communications Inc. (SPCI) to develop a UT SBB Safety Test tool, and associated test scripts to address the first requirement above. The functionality of this tool is being implemented using a phased approach and is expected to evolve over the medium/long term towards a fully automated UT SBB Safety Test Tool that can verify all of the functionality of the ACARS Aircraft Gateway (AAGW). The full functionality of the AAGW includes:

- Selection and management of the satellite link (SBB or Classic Aero) including log on/off and failover as required;
- Transport of air-to-ground and ground-to-air ACARS messages via the selected satellite link;
- In the case of SBB, interfacing with its ground-side peer, the ACARS Ground Gateway (AGGW), using the interface specified in section 3.13 of [1].

The use cases to be tested are well summarized in section 3.11 of [1]. Tests to verify the formatting and handling of each field of the AAGW/AGGW messages, as well as handling of incorrect or inopportune message contents, are included in the use case tests to the extent possible with additional tests defined only when it would have been awkward or impossible otherwise.

1.2 Scope

The test tool currently supports three modes of operation, referred to as Phases 1 to 3.

In the Phase 1 scenario, AIGI verification is accomplished using a scripting engine that exercises the following over an IP link:

- AIGI protocol (normal operation)
- AIGI protocol with error injection and failure conditions
- Verification of AIGI message content and format, including unformatted/incorrect messages
- ACARS message transfer

Phase 2 adds support for an ARINC-429 interface to the SDU. Phase 3 adds support for operating the scripts through the real air interface, e.g., on the bench using a BPLT and BGAN Network Emulator (BNE) or equivalent.

This document describes the Phase 1, 2 and 3 test configurations, how to install and configure the test tool for each configuration, how to install and configure the test scripts for each configuration, and how to run the test scripts and evaluate the results.

1.3 Conventions

Hexadecimal values are shown with a leading dollar sign: e.g., $FF = 255 decimal.

1.4 Applicable Documents

The following documents are applicable to the extent that they are referenced herein.

[1] *BGAN SDD: Volume 3 Chapter 2: Swift Broadband Oceanic Safety Service,* Chapter Revision 1.3, January 2013, Inmarsat.

[2] *Mark 33 Digital Information Transfer System (DITS) Part 1: Functional Description, Electrical Interface, Label Assignments and Word Formats*, ARINC Specification 429 Part 1-17, May 17, 2004, Aeronautical Radio, Inc., www.arinc.com .

[3] *Aviation Satellite Communication System Part 2: System Design and Equipment Functional Description*, ARINC Characteristic 741P2-8, May 31, 2006, Aeronautical Radio, Inc., www.arinc.com .

[4] *Domain Names – Concepts and Facilities,* Network Working Group RFC-1034, November 1987.

[5] *Domain Names – Implementation and Specification,* Network Working Group RFC-1035, November 1987.

[6] *BGAN Network Emulator User's Manual,* Version 1.34, 7 October 2011, Gatehouse.

[7] *SBB Safety AAGW Test Case Descriptions*, SPCI document DC-350201.

[8] *SBB Safety Test Tool Interface Specification*, SPCI document DC-350202.

[9] *SBB Safety Test Tool User's Guide*, SPCI document DC-350301.

[10] *SBB Safety Test Tool Design Description*, SPCI document DC-350401.

[11] *SBB Safety Test Case Detailed Design Descriptions*, SPCI document DC-350402.

[12] *SBB Safety AAGW Emulator Design Description*, SPCI document DC-350403.

[13] *ARINC-429 Commentary*, Version 2.1, 10 May 1999, SBS Avionics Technologies.

[14] *BPLT Operator's Reference Manual,* SPCI document DC-210301.

[15] *BPLT Script Language Manual,* SPCI document DC-210302.

[16] *BPLT User's Guide,* SPCI document DC-210303.

[17] *PLT-H Hardware Manual,* SPCI document DC-210304.

[18] *CEI-x20-SW Installation Guide for the CEI-220, CEI-420/420A, CEI-520/520A, CEI-620, and CEI-820/820TX,* GE Intelligent Platforms Inc., www.ge-ip.com .

[19] *CEI-100/CEI-200/CEI-x20 User's Manual,* GE Intelligent Platforms Inc., www.ge-ip.com .

1.5 Acronyms

The following acronyms may be used herein:

AAGW	ACARS Aircraft Gateway
ACARS	Aircraft Communications And Reporting System
AES	Aircraft Earth Station
AGGW	ACARS Ground Gateway
AIGI	ACARS Intra-Gateway Interface
ARINC	Aeronautical Radio Inc.
BGAN	Broadband Global Area Network
BNE	BGAN Network Emulator
BPLT	BGAN Physical Layer Tester
BSCR	BPLT Scripting Tool
CMU	Communications Management Unit
CN	Core Network
CSP	Communications Service Provider
CU	Channel Unit
DNS	Domain Name System
DP	Distribution Partner
GDU	Ground Data Unit
GES	Ground Earth Station
IRS	Inertial Reference System
IUT	Implementation Under Test
IP	Internet Protocol
MU	Management Unit
PLT	Physical Layer Tester
SBB	SwiftBroadband
RAN	Radio Access Node
RFU	Radio Frequency Unit
SDL	Specification and Description Language
SDU	Satellite Data Unit
SPCI	Square Peg Communications Inc.
TBC	To Be Confirmed
TBD	To Be Determined
TCP	Transport Control Protocol
UDP	User Datagram Protocol
UT	User Terminal

2. ARCHITECTURE

2.1 Test Controller

2.1.1 Overview

The Test Controller drives the protocol conformance testing and the collation of the results. It also implements the physical interfaces to the implementation under test, including Ethernet and, for Phase 2/3, ARINC-429.

Table 2-1 summarizes the key elements of the Test Controller.

Table 2-1 Test Controller Functions

Function	Description
Script engine	Compiles and executes scripts written in SPCI's proprietary PLT script language. Scripts may be executed individually or grouped into a test suite.
Test suite log	Logs the Pass/Fail/Inconclusive result for each test script in a test suite. Provides the overall summary of test suite execution. Logs are plain text ASCII.
Test case log	Provides detailed logging for each individual test script, including all messages exchanged between the Test Controller and the Implementation Under Test. Logs are plain text ASCII.
Event log	Provides detailed logging at a system level; i.e., independent of script execution. Includes facilities for logging LAN messages and ARINC-429. Useful for debugging system issues. Logs are plain text ASCII.
Ethernet interface	Provides facilities for the transmission and reception of TCP streams and/or UDP packets on selected ports.
ARINC-429 interface	Provides facilities for the transmission and reception of ARINC-429 labels, and when used with a GE Fanuc/Condor CEI-520A card, for the transmission and reception of files using the Williamsburg protocol.

The Test Controller implements the same Windows-based user interface as is used in the BPLT. It provides an integrated environment for script development and execution. Details on the BPLT's operator interface can be found in [14].

2.1.2 Platform

For Phase 1 or 2, the Test Controller can be implemented on a PC (running Windows XP or Windows 7) or on a BPLT. In the former case, a derivative of the BPLT software, BSCR, which supports the required scripting functionality is utilized. In the latter case, the standard BPLT application software is used.

For Phase 3, the Test Controller is expected to be implemented on a BPLT since the BPLT will serve as the physical layer when testing via the air interface.

2.1.3 Script Language

The script language has the following general capabilities:

- sending and receiving data via a serial port, LAN messages (TCP or UDP), or an IEEE-488 or ARINC-429 interface to control and monitor external equipment;
- prompting the user to perform actions;
- reading and writing files;
- logging and displaying events and results.

The script programming language is conceptually similar to the common procedural languages such as Basic, Pascal, and C. It provides looping and conditional structures, infix expression evaluation, user-defined variables and procedures, compile-time symbol substitution (similar to `#define` in C), and conditional compilation (similar to `#ifdef` in C). Inclusion of other source files (similar to `#include` in C) is also supported to allow common definitions or procedures to be reused.

The primary differences from conventional languages are:

- there is no explicit statement terminator (by comparison, Basic uses end-of-line, while Pascal and C use semicolon).
- only scalar variables are supported directly (not arrays or structures), although facilities exist to implement these through the equivalent of pointers.
- all variables are static and are globally accessible.
- expression evaluation is strictly left-to-right, except as modified by parentheses (all binary operators have the same precedence).
- built-in functions are provided that support the requirements of the PLT application.

Only one script file can be run at a time. However, a list of script files can be specified in a “test suite” file and run in sequence. Logging facilities are provided to produce a record of script and test suite execution.

Script and test suite source files are plain ASCII text and can be created with any text editor (e.g., Windows Notepad). A text editor is also integrated with the script file control facility.

Scripts can be saved in compiled format. Compiled scripts begin execution more quickly than uncompiled scripts when run.

Detailed information on the Tester script language can be found in [15].

2.2 Phase 1 Test Architecture

Figure 2-1 illustrates the test configuration employed in Phase 1. The PC or PLT hosts the BPLT-based scripting engine and has an Ethernet connection to either the AAGW or an AAGW simulator running on a PC. The IRS and CMU messaging are simulated across the Ethernet.

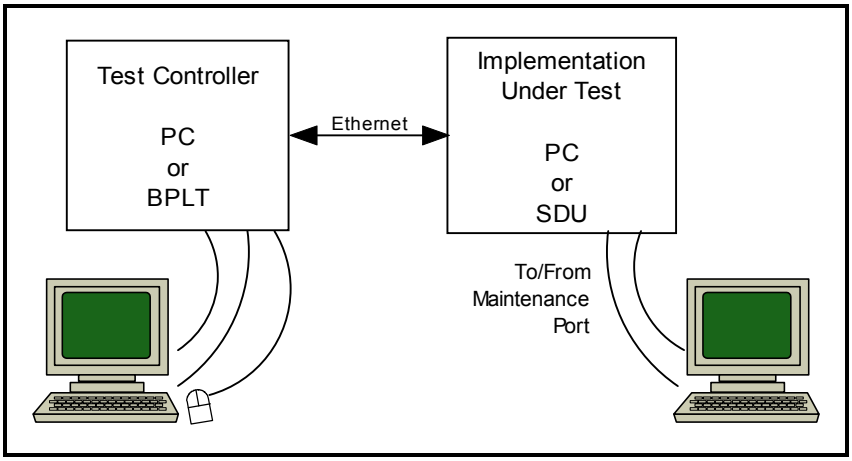


Figure 2-1 Phase 1 Test Configuration

2.3 Phase 2 Test Architecture

Figure 2-2 illustrates the test configuration employed in Phase 2, where the test engine directly interfaces with the CMU and the IRS functions on the SDU via ARINC-429 rather than simulating these interfaces through the Ethernet. For Phase 2 operation, the test controller must incorporate a GE Fanuc/Condor CEI-520A ARINC-429 interface card, which allows the BPLT application to interface to the SDU via ARINC-429 / Williamsburg.[1]

In Phase 2, tests are still run using an IP connection to the SDU/AAGW, but the ARINC-429 interfaces are now implemented by the test controller, allowing it to send and receive ACARS messages and assert/check NOCOMM or other discretes without operator intervention or IP messaging. The control and modem interfaces continue to be simulated via IP.

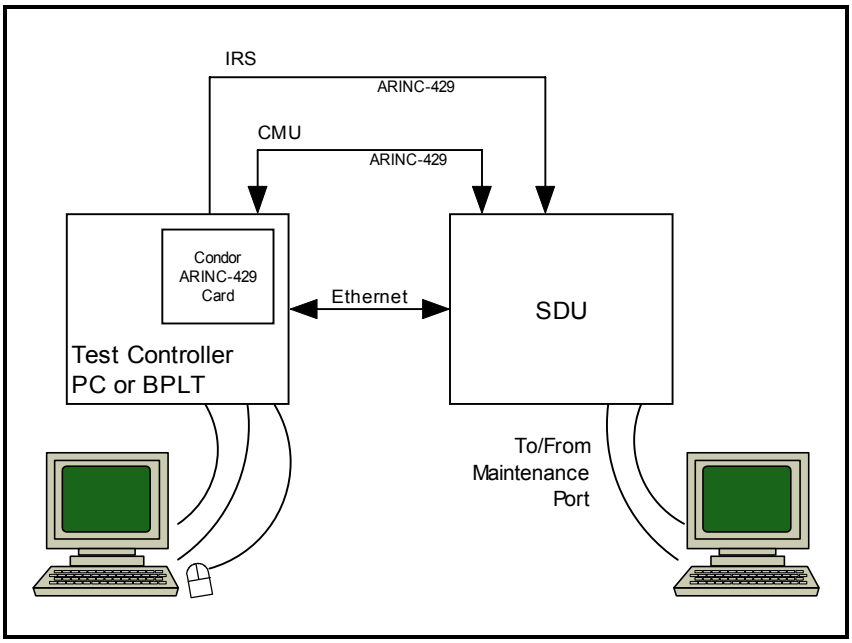


Figure 2-2 Phase 2 Test Configuration

[1] The Ballard OmnibusBox cannot be used since Williamsburg is not supported using this device. The CEI-530 is also not supported at this time.

2.4 Phase 3 Test Architecture

Figure 2-3 illustrates the test configuration employed in Phase 3, where the AIGI and DNS traffic is carried over the air interface rather than over Ethernet. The scripting engine now executes as part of the BPLT application running on a PLT-H platform which incorporates a GE Fanuc/Condor CEI-520A ARINC-429 interface card.

In SPCI's verification setup, the Gatehouse BGAN Network Emulator (BNE) is used to simulate the RAN and Core Network (CN), which the SBB modem interfaces to at RF via the BPLT. AIGI messages and DNS queries and responses are carried as UDP packets which are presented to or sourced from the BPLT via a UDP interface.

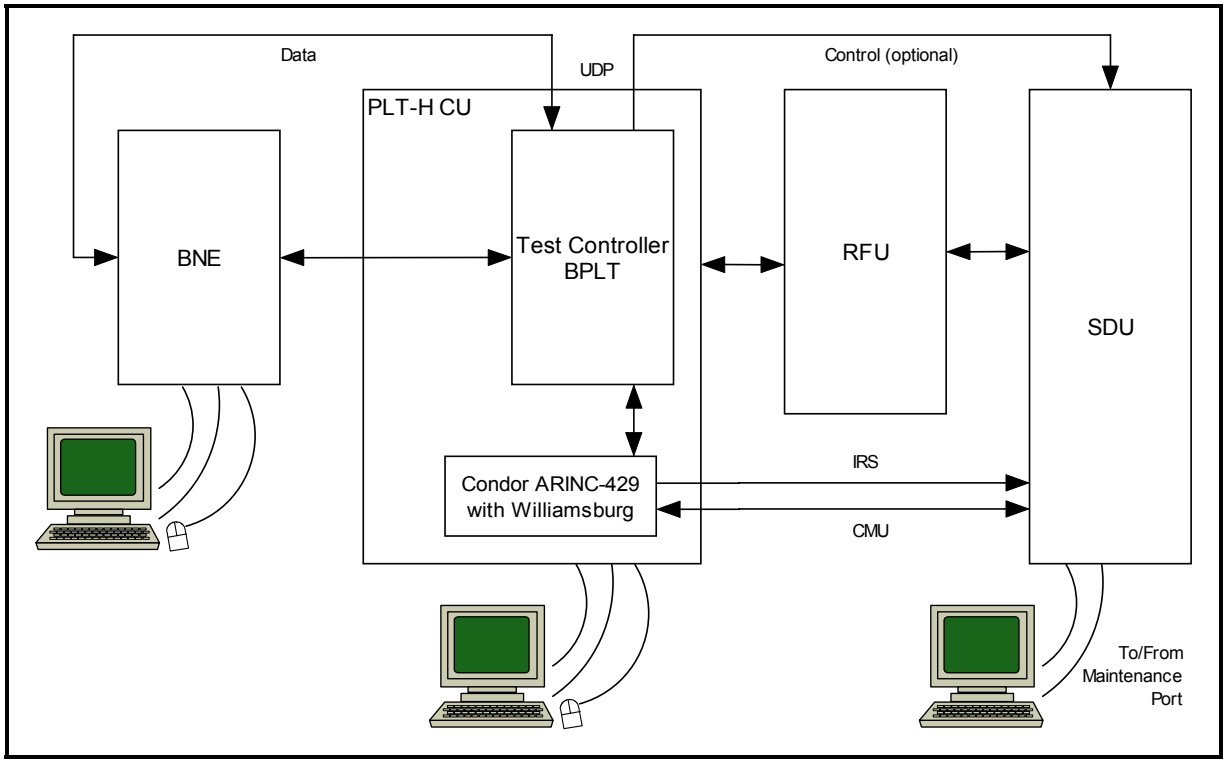


Figure 2-3 Phase 3 Test Configuration

In Phase 3, the BNE is not controlled by the BPLT, hence the test coverage is restricted to that of Phases 1 and 2. The Control interface can be implemented via IP (as in Phases 1 and 2) or via prompting the operator.

3. INSTALLATION

3.1 System Requirements

Phase 1

PC with:

- 2 GHz or faster x86 (32-bit) processor with SSE2 instruction set
- 1 GB RAM
- Windows XP or Windows 7
- C: and D: hard drives (see section 3.1.1)

Phase 2

As per Phase 1, plus:

- Condor CEI-520A with at least 2 transmit channels and 2 receive channels

Phase 3

SPCI PLT-H running BPLT 6.02 or later with:

- Condor CEI-520A with at least 2 transmit channels and 2 receive channels

3.1.1 D: Drive

The Tester software and all other distribution files must reside on drive C:, while user configuration data and log files must reside on drive D: .

If the PC on which the Tester is to be run does not have a D: drive, then one must be created prior to installing the software. There are two methods by which this can be done:

1. Create a D: partition (preferred method)

   A commercial program such as EaseUS® Partition Master (see http://www.partition-tool.com/professional.htm) may be used to repartition the PC's hard drive and create a D: partition. The D: drive should have at least 200 MB of free space.

Re-partitioning a drive is at the user's risk. Be sure to back up all critical information prior to re-partitioning.

2. Create a logical D: drive as a directory on the C: drive (alternate method)

   With this method, the Windows/DOS subst command is used to associate a path with a drive letter. While simple, care must be taken to avoid conflicts with an existing drive D: (e.g., a CD-ROM drive) or with USB drives. The latter can be particularly problematic since inserting a new USB key can cause Windows to use drive D: even if it is already used by a subst command.

   Information on using the subst command can be obtained from various online resources. Google also offers a psubst batch file which makes the substitution permanent (see http://code.google.com/p/psubst/).

   If D: is already used (e.g., by a CD-ROM drive, or periodically by a USB drive), before performing the drive substitution, the Windows Disk Management tool can be used to reassign the drive letter; e.g., so that the CD-ROM is drive E: . To do this, run

compmgmt.msc from the Windows Start Menu, select Disk management in the left hand pane and then reassign the drive letters as required.

3.2 Hardware Installation

3.2.1 Installation Of CEI-520A Condor ARINC 429 Card

Consult the CEI-520A Quick Start guide [18] and CEI-520A User's Manual [19] for installation instructions.

The CEI-520A provides a 68-pin SCSI-3 connector (AMP/Tyco P/N 787171-1) which carries the ARINC 429 signals for each channel. The pinout of this connector can be found in [19], CEI-520/520A ARINC 429 I/O Connector (P2). A SCSI-3 male to SCSI-3 male adapter cable (GE P/N CONSCSI3-6) is also available, and can be user-modified to interface SCSI-3 to other connector types.

3.2.2 LAN Interface

The Test Controller requires at least one LAN interface, with a lossless connection to the IUT. This can be via a direct connection or via a switch.

The SBB test scripts require that the Tester have at least two IP addresses (on the same subnet). To set up additional IP addresses on a Windows PC:

1. Go to Control Panel/Network Connections/Local Area Connection.
2. Click on Properties.
3. Scroll down and select TCP/IP and then select properties.
4. Click on Advanced and then click on Add.
5. Add the desired extra IP address(es).

3.3 Software And Script Installation

3.3.1 Test Controller Software

If a BPLT is serving as the Test Controller (on a PLT-H or PLT-L CU), follow the instructions given in section 8 of the BPLT User's Guide [16].

If the Test Controller is being hosted on a PC, follow the procedure given in Table 3-1.

Table 3-1 BSCR Software Update

Action	Method
Back up or rename **C:\Plt\Bplt** and **C:\Bplt_Dat** . Do not delete or rename **C:\Plt\Common**.	
Copy the SBB release onto the Test Controller.	Copy **BSCR_mnn.b.exe** from the distribution CD or as downloaded from the SPCI web site to a directory such as c:\Plt_install. Note: mnn.b is the release version number; for example, Bscr_603.2.exe is the second build of Bscr 6.03 .
Install the Test Controller software.	Run **BSCR_mnn.b.exe** from the distribution CD or as downloaded from the SPCI web site.
Other	Perform any other actions called for by the release notes accompanying the update.

3.3.2 Test Scripts

The SBB test scripts are distributed as an executable installer. Follow the procedure given in Table 3-2 to install or update the SBB scripts.

Table 3-2 SBB Script Installation

Action	Method
If this is an update installation, back up or rename **C:\Bplt_dat\SBB** . Do not delete or rename any other directories under **C:\Bplt_dat**.	
Copy the SBB release onto the Test Controller.	Copy **SbbTest_mnn.b.exe** from the distribution CD or as downloaded from the SPCI web site to a directory such as c:\Plt_install. Note: mnn.b is the release version number; for example, SbbTest_302.1.exe is the first build of SbbTest V3.02 .
Install the test scripts.	Run **SbbTest_mnn.b.exe** and follow any instructions. The SBB scripts and documentation will be installed into C:\Bplt_dat\SBB. The directories under SBB are: .\Doc - documentation .\Inc - common include files .\Ini - common initialization files .\Sui - suite files used to compile or run many files in sequence .\SBB0nn – test scripts pertaining to a particular use case .\Test – miscellaneous files used for whitebox testing
Compile the test scripts	Start the test controller software. Open the **Script** window and navigate to C:\Bplt_Dat\SBB\Suite. Select RunAll.sui and click **Compile**.
Other	Perform any other actions called for by the release notes accompanying the update.

3.4 Initialization Files

After installation, the Test Controller and scripts must be configured for the particular test environment required. Configuration information is stored in initialization (.ini) files. This allows for easy customization of the test configuration and script behaviour without requiring modifications to the scripts themselves.

As shipped, the software and script releases already contain the majority of configuration information required. All configurable parameters are contained in initialization files, which are modeled on a Windows `.ini` file. Each file consists of a series of sections identified with a section name contained in square brackets, e.g.

```
[Arinc429]
```

Under each section name is a series of parameters defined in the form

```
ParameterName=ParameterValue
```

The value may be a string, a numeric or other value depending upon its application. Within a `.ini` file, comments may be included by prefacing a line with a semicolon “;” character. The supplied `.ini` files include comments that explain how each parameter is used and the valid range of values.

Some parameters may be optional and a null value is specified with a line of the form

`ParameterName=`

with no value specified.

The files stored in the C:\ directory tree are read-only, configuration controlled files. If a user desires to modify a setting contained in an initialization file, this is accomplished by creating a separate file on drive D: .

For Test Controller (BPLT) ini files, the user parameters are specified in files in D:\Bplt_dat\Ini. For example, to update a parameter specified in C:\Plt\Bplt\Cu\Car\Car_ex.ini, a file called Car_ex.ini is created in D:\Bplt_dat\Ini. A parameter setting in an update file on the D: drive will override that specified on the C: drive.

For test script ini files, the user parameters are specified in D:\Bplt_dat\SBB\Ini. For example, to update a parameter specified in C:\Bplt_dat\SBB\Ini\SBB_TestPar.ini, a file called SBB_TestPar.ini is created in D:\Bplt_dat\SBB\Ini. Again, a parameter setting in an update file on the D: drive will override that specified on the C: drive.

An update file need only include the sections and properties that are different from the original file. Other sections and properties are taken from the C:\ copy of the file. For parameters read from SBB_TestPar.ini, the testcase log will indicate the value used, and the source of that value, with “(User)” indicating that it came from the D: drive.

CAUTION

No files should be created or modified in directory C:\Plt\Bplt or its subdirectories, or in C:\Bplt_dat\SBB or its subdirectories, as these directories will be completely replaced if an updated version of the software is installed.

3.4.1 Test Controller Initialization Files

The Test Controller uses the standard BPLT initialization files. For Phase 2 and 3, where ARINC-429 communication is required, the BPLT must be configured to use a Condor CEI-520A with at least two channels.

The following entries should be made in D:\Bplt_dat\Ini\Car_Ex.ini:

```
[Board]
Type=CEI520a  // Board type (case-insensitive ): CEI520a or OmniBusBox
BoardNum=0    // Board or card number
CoreNum=0     // May need to specify CoreNum=1 for OmniBusBox with two cores installed
PollingIntervalMs=40
```

The following entries should be made in D:\Bplt_dat\Ini\Cin.ini:

```
[Tasks]
Car_Ex2=C:\Plt\Bplt\Cu\Car\Car_Ex\Car_Ex.exe C:\Plt\Bplt\Cu\Car\Car_Ex\Car_Ex2.ini
```

Other BPLT ini file parameters may be set, if required, as described in [16].

3.4.2 Test Script Initialization Files

When an SBB test script is executed, it reads from SBB_TestPar.ini all of the values that it requires on startup. If a value is not specified, or is invalid, the script will terminate and report an error.

Section 5.2 describes the parameters specified in SBB_TestPar.ini .

4. TEST SCRIPT COVERAGE AND ORGANIZATION

4.1 Test Coverage

Phase 1/2/3 test coverage is restricted to scenarios that can be verified using an IP connection to the Implementation Under Test (IUT), including:

a) Validation of the all fields, including transaction id, of all AIGI messages sent by the terminal (AAGW to AGGW).

b) Validation of the behaviour of the AAGW under the use case tests specified in Table 4-1.

c) Validation that the AAGW handles configuration parameters properly.

For convenience, the test cases are divided into a number of test groups. The test groups and high level test case descriptions are documented in [7] while the detailed test steps are documented in [11].

Table 4-1 summarizes the Phase 1/2/3 testcases. Italicized test groups are not implemented in Phase 1/2/3.

Table 4-1 Testcase Summary For Phase 1/2/3 Configuration

Test Group	Description	SDM [1] Reference
SBB001	AAGW establishes best available link with ground	UC1
SBB002	AAGW logs in to SBB ACARS service	UC2
	Ground To Air keep alive	UC3
	AAGW receives invalid message	UC4
	Ground To Air Configuration Message	UC23
	Format: ac_logon_rq, ac_logon_rq_n	
	Format: ac_keepalive_ack	
SBB003	AAGW logs out of SBB ACARS service	UC5
	Format: ac_logoff_rq, ac_logoff_rq_n	
	Format: ac_acars_msg, ac_acars_msg_n	
	Format: ac_acars_ack, ac_acars_ack_n	
SBB005	Air to ground ACARS message	UC6
	Format: ac_acars_msg, ac_acars_msg_n	
SBB006	Ground to air ACARS message	UC7
	Format: ac_acars_ack, ac_acars_ack_n	
SBB007	*Satellite hand-over*	*UC8*
SBB008	*Out of SBB coverage*	*UC9*
SBB009	AGGW logs out aircraft due to return link inactivity	UC10
SBB010	AAGW detects unexpected PDP context disconnect	UC11
SBB011	*AAGW detects loss of synchronization to SBB forward bearer*	*UC12*

Test Group	Description	SDM [1] Reference
SBB012	AAGW detects forward link inactivity	UC13
	Format: ac_keepalive	
SBB013	CMU asserts ACARS NOCOMM	UC14
SBB014	AGGW detects failure of CSP link	UC15
SBB015	AGGW receives message from aircraft in logged off state	UC16
SBB016	*AGGW receives log-on request message from aircraft in logged on state*	*UC17* *Applicable to AGGW only*
SBB017	*Transition to higher preference service after timeout*	*UC18*
SBB018	*Transition to higher preference service after entering coverage region*	*UC19*
SBB019	AAGW receives message from unexpected IP address	UC20
SBB020	*AAGW Receives Invalid Message*	*UC 20 is tested in various other testcases*
SBB021	CSP Ground To Air Ping	UC21
	Format: ac_csp_ping_ack	
	Format: ac_msg_nak, ac_msg_nak_n	Also tested in other test groups
SBB022	AIGI Test Message	UC22
	Format: ac_test_ack, ac_test_ack_n	
SBB024	*AGGW receives message from aircraft when not serving an ocean region*	*UC23* *Applicable to AGGW only*
SBB025	*AGGW receives invalid message from aircraft*	*UC24* *Applicable to AGGW only*

Most of the test suite must be run twice, once with position reporting disabled and once with position reporting enabled.

4.2 Directory Structure

Table 4-2 illustrates the directory structure used by the SBB Safety test scripts.

Table 4-2 Directory Structure

Directory	Function
C:\Bplt_Dat\SBB	SBB project directory
Doc	Documentation
Inc	Common include files
Ini	Initialization files
SBB001 .. SBB022	Test group specific scripts and include files
Suite	Test suite file and any associated utility scripts
Test	Test scripts used for verifying SBB Safety test scripts
D:\Bplt_Dat\SBB	
Ini	User overrides for initialization files
D:\Bplt_Log	Location of Event Log and Test Suite log files
SBB	Default location for Test Case log files (can be overridden in ini file)

5. TEST SCRIPT INTERFACES AND CONFIGURATION

5.1 Interfaces

The following sections specify the interfaces between the test script and the Implementation Under Test (IUT). For Phase 1, each interface is implemented on a separate UDP port. The actual port numbers are defined in a configuration file. In subsequent phases, some interfaces are implemented over ARINC-429.

Each message is encapsulated in a single UDP packet, and only one message may be present in a packet. In all cases, the UDP checksum field is used to determine the packet's validity; i.e., there is no additional checksum.

5.1.1 IP Interface Overview

Tester/AAGW messages are transported between the Tester and the IUT via UDP, with different ports used for different interfaces. Use of UDP requires a lossless connection between the Tester and the unit under test.

Table 5-1 lists the ports expected to be used. All IP/ports used on a given entity (Tester or IUT) should be non-overlapping so that the UDP ports can be created.

Table 5-1 UDP Ports

Interface	Use	Default Port At IUT	Default Port At Tester	Notes
Control	Messages to AAGW from higher level entities in the SDU.	60001	60011	Tester may be configured to prompt the operator rather than utilizing IP.
Williamsburg	Messages that would normally be carried over ARINC 429 / Williamsburg to/from the CMU/MU.	60002	60012	In Phases 2 and later, Tester may be configured to utilize ARINC 429 rather than IP.
Status	SDU to/from MU/CMU status messages that would normally be carried over ARINC 429.	60003	60013	In Phases 2 and later, Tester may be configured to utilize ARINC 429 rather than IP. In this case, these messages are carried over the same physical interface as the messages transmitted over the Williamsburg interface.
IRS	IRS messages that would normally be carried over ARINC 429.	60004	60014	In Phases 2 and later, Tester may be configured to utilize ARINC 429 rather than IP.

Interface	Use	Default Port At IUT	Default Port At Tester	Notes
AIGI	AIGI messages encapsulated in UDP packets exchanged between AAGW and AGGW that would normally be carried over the air interface.	Source 30001 Dest. 30000	Source Random Dest. 30000	From the perspective of the IUT, it is only communicating with one AGGW at a time, even though the Tester may emulate more than one. **From the perspective of the Tester, two separate ports are created, one for the primary AGGW IP address and one for the alternate AGGW IP address.** In Phase 3, when the SDU is operating over the air interface, the Tester will exchange AIGI packets with the network emulator rather than with the IUT's test interface.
DNS	DNS queries and responses encapsulated in UDP packets exchanged between AAGW and a DNS server via the air interface.	N/A (will be sent to the source port of the request)	53	From the perspective of the IUT, it is only communicating with one DNS server at a time, even though the Tester may emulate more than one. **From the perspective of the Tester, two separate ports are created, one for the primary DNS server IP address and one for the secondary DNS server IP address.** In Phase 3, when the SDU is operating over the air interface, the Tester will exchange DNS packets with the network emulator rather than with the IUT's test interface.

The test script configuration files allow the IP addresses and ports to be specified for each interface, with the following exceptions:

1. The IP address of the AAGW is determined from the ac_logon_rq or ac_logon_rq_n packet.
2. DNS replies are sent to the source IP address and port of the DNS query.

The primary AGGW IP address (as specified by [Aigi]PrimaryAggwIpAddress, see Table 5-2) is the main IP address of the Tester and is used for the Control and AIGI interfaces. The user may set an alternative IP address, [Arinc429]LocalIpAddress to be used for the Williamsburg, Status and IRS interfaces In a Phase 3 configuration where IP is used for ARINC 429, this can be used to avoid sending Tester-bound Williamsburg and Status packets over the air interface.

5.1.2 Control Interface

The messages carried over the UDP *Control* port are specified in [8].

A configuration parameter determines whether this interface is implemented via IP or instead the operator is prompted for each Control message.

A configuration parameter determines whether informational messages related to test progress are sent to the IUT.

A *Restart AAGW* message is sent at the start of every test script that needs to force a new AAGW log on. Scripts which do not require a new log on should only issue a *Restart AAGW* if more than ac_t1 + ac_r1 * ac_t2 has elapsed between testcases; i.e., the Tester assumes that the AAGW is logged out due to inactivity if more than ac_t1 + ac_r1 * ac_t2 has elapsed.

5.1.3 Williamsburg Interface

The messages carried over the UDP *Williamsburg* port are specified in [8].

5.1.4 IRS Interface

The messages carried over the UDP *IRS* port are specified in [8].

5.1.5 ARINC Status Interface

The messages carried over the UDP *Status* port are specified in [8].

For *SDU to ACARS MU/CMU Status* messages, the Tester verifies the contents of the label (bits 1-8), NOCOMM (bit 11), log-on (bit 17), SSM (bits 30-31) and parity (bit 32) fields.

For *SDU to ACARS MU/CMU Join/Leave* messages, the Tester verifies the contents of the label (bits 1-8), GES ID (bits 9-16), satellite ID (bits 17-22), service type (bits 23-25), NOCOMM (bit 29), SSM (bits 30-31) and parity (bit 32) fields. For *Join* messages, the GES ID must match that provided in the gw_logon_rp and the satellite ID and service type must match the corresponding configuration parameters sent to the IUT in the *Restart AAGW* message. For *Leave* messages, the GES ID and satellite ID fields are not checked.

5.1.6 Modem Interface

The messages carried over the UDP *AIGI* port or *DNS* port are specified in [8].

DNS queries and responses are formatted as described in [5]. If the DNS query is invalid, the script will fail; i.e., it will not respond with a DNS failure.

5.2 Configuration Parameters

The test scripts are configurable via a Windows format ini file. Table 5-2 lists the ini file entries.

Some parameters are checked for validity on startup. If a parameter is not specified, the default value is used. When no default value is shown, initialization will be terminated if a valid value is not specified.

Table 5-2 Test Script Configuration Parameters

Section	Parameter	Values/Notes
[Control]	TransportOverIp	True: Control messages are transported over IP False: The operator is prompted to initiate the action. Default is TRUE
	IutControlIpAddress	IP address of the IUT; i.e., IP address to which to send control messages from the script. Only used if IP transport is selected. This parameter is not defined by default.
	IutControlPort	UDP port number (at the IUT) used to receive control messages from the script; i.e., destination for script-originated packets. Only used if IP transport is selected. Default 60001.
	LocalControlPort	UDP port number (at the Tester) used to receive control messages from the IUT; i.e., destination for AAGW-originated control packets (not currently applicable). Also used as the source port number in control messages sent to the UT. Only used if IP transport is selected. Default 60011.
	ReportTestEvents	True: Test events are reported over the Control interface. False: Test events are not reported over the Control interface. Only used if IP transport is selected. Default is TRUE.
	ReportTestProgress	True: Test start and completion are reported over the Control interface. False: Test start and completion are not reported over the Control interface. Only used if IP transport is selected. Default is TRUE.
[Arinc429]	TransportOverIp	True: ARINC messages are transported over IP False: ARINC messages are sent via a physical ARINC-429 interface Only True is valid for the Phase 1 configuration. Default is TRUE.

Section	Parameter	Values/Notes
	IutArincIpAddress	IP address of the IUT; i.e., IP address to which to send ARINC 429 messages from the script. Only used if IP transport is selected. This parameter is not defined by default.
	IutWilliamsburgPort	UDP port number (at the IUT) used to receive Williamsburg messages from the script; i.e., the destination UDP port number for Williamsburg messages sent by the script. Only used if IP transport is selected. Default 60002.
	IutStatusPort	UDP port number (at the IUT) used to receive Status messages from the script; i.e., the destination UDP port number for Status messages sent by the script. Only used if IP transport is selected. Default 60003.
	IutIrsPort	UDP port number (at the IUT) used to receive IRS messages from the script; i.e., the destination UDP port number for IRS messages sent by the script. Only used if IP transport is selected. Default 60004.
	LocalIpAddress	Local IP address used for Williamsburg, Status and Irs ports on the Tester. For Phase 1, can be the same as [Aigi]PrimaryAggwIpAddress but if Phase 3 is run with [Arinc429]TransportOverIp = TRUE then it should be set to an alternate address; e.g., [Aigi]AlternateAggwIpAddress. This parameter is not defined by default.
	LocalWilliamsburgPort	UDP port number (at the Tester) used to receive Williamsburg messages from the IUT; i.e., destination for AAGW-originated Williamsburg packets. Also used as the source port number in Williamsburg messages sent to the IUT. Only used if IP transport is selected. Default 60012.
	LocalStatusPort	UDP port number (at the Tester) used to receive Status messages from the IUT; i.e., destination for AAGW-originated Status packets. Also used as the source port number in Status messages sent to the IUT. Only used if IP transport is selected. Default 60013.

Section	Parameter	Values/Notes
	LocalIrsPort	UDP port number (at the Tester) used to as the source port number in IRS messages sent to the IUT. Only used if IP transport is selected. Default 60014.
	IrsChannel	ARINC 429 logical transmit channel for IRS labels transmitted from the Tester. Must be different than CmuChannel. Only used if IP transport is not selected. Default is 2
	IrsBusSpeed	Low or High Only used if IP transport is not selected. Default is 'HIGH'
	IrsDelayMs	Delay to wait after sending IRS data over ARINC 429. Default is 1000 ms.
	CmuChannel	ARINC 429 logical transmit channel for CMU label 270 and Williamsburg transmitted from the tester. ARINC 429 logical receive channel for SDU label 270 and Williamsburg received by the Tester. Must be different than IrsChannel. Only used if IP transport is not selected. Default is 1
	CmuBusSpeed	Low or High Only used if IP transport is not selected. Default is 'LOW'
	CmuLocalSal	Octal 304 => ACARS/CMU Only used if IP transport is not selected. Default is $C4
	CmuRemoteSal	Octal 307 => SDU#1 Only used if IP transport is not selected. Default is $C7
	CmuDestinationCode	ASCII 'S' => SDU Only used if IP transport is not selected. Default is $53
	DisconnectWbpAtStartOfEachTest	Set TRUE to force a Williamsburg ALO/ALR on the CMU interface at the start of each test. If FALSE, the script will only issue ALO/ALR if it believes that the Williamsburg connection is not already up. Default is TRUE.

Section	Parameter	Values/Notes
[Aigi]	PrimaryAggwIpAddress	Default IP address of the AGGW (Tester). Provided to the AAGW in the DNS response. Used for all tests except for those which test the AAGW's handling of AGGW site changes, where the AGGW's IP address changes. This parameter is not defined by default.
	AlternateAggwIpAddress	Alternate IP address of the AGGW (Tester). Provided to the AAGW in the DNS response in tests which verify the AAGW's handling of AGGW site changes. This parameter is not defined by default.
	AagwSrcPort	Source UDP port number in AIGI messages sent from the IUT; i.e., AAGW sends AIGI messages from this UDP port. Default 30001 (per [1] 3.13.1).
	AagwDestPort	UDP port number at the IUT used to receive AIGI messages; i.e., AAGW receives AIGI messages from this UDP port. Default 30000 (per [1] 3.13.1).
	MinAggwSrcPort	Minimum and maximum UDP port numbers from which AIGI messages may be sent by the Tester; i.e., the Tester AGGW may send AIGI messages from any local UDP port within MinAggwSrcPort ≤ port ≤ MaxAggwSrcPort. Default 30002 … 31000.
	MaxAggwSrcPort	
	AggwDestPort	UDP port number at the Tester used to receive AIGI messages; i.e., AGGW receives AIGI messages from this local UDP port. Default 30000 (per [1] 3.13.1).
	TimestampToleranceMs	Number of ms of tolerance between expected timestamp and reported timestamp. Default 5000. Set to 0 to indicate that absolute timestamp values should not be checked.
[Dns]	AcarsHostName	Expected ACARS host name in DNS lookups. Default 'aggw.sbb.inmarsat.com'.
	PrimaryDnsIpAddress	IP address of primary DNS server. Default 0.0.0.0 which means the IP address of the Tester. **If operating over the air interface, this is the IP address provided for the PDP context that is set up by the AAGW. This must be predictable.**

Section	Parameter	Values/Notes
	SecondaryDnsIpAddress	IP address of secondary DNS server. Default 0.0.0.0 which means the IP address of the Tester. **If operating over the air interface, this is the IP address provided for the PDP context that is set up by the AAGW. This must be predictable.**
	LocalDnsPort	UDP port number (at the Tester) used to receive DNS queries from the IUT. Also used as the source port number in DNS responses sent to the IUT. Default 53.
[Sdu]	IcaoAddress	24-bit ICAO address (AES ID) Default is $010203
	ReportLocation	True or False True if location information is reported in AIGI messages; FALSE otherwise. Default is TRUE
	LocationSource	If ReportLocation is TRUE, indicates source of the location report. 0 = Inertial Reference System (IRS) 1 = Global Positioning System (GPS) 2 = Hybrid 3 = Reserved Default is 0.
	ProtocolVersion	Per [1] 3.13.2.5.1 Default is 1
	SatId	Per [1] 3.13.2.5.1 Default is 1
	SpotId	Per [1] 3.13.2.5.1 Default is 0
	Imsi	Per [1] 3.13.2.5.1 Specified as a string (0-15 digits). Script will pad with digit 15 ($F) to maximum length as required. Default is '123456789012345'
	ImeiSvn	Per [1] 3.13.2.5.1 Specified as a string (0-16 digits). Script will pad with digit 15 ($F) to maximum length as required. Default is '8900123456789012'
	TerminalType	Per [1] 3.13.2.5.1 Default is 1

Section	Parameter	Values/Notes
	TypeApprovalId	Per [1] 3.13.2.5.1 Specified as a string (0-6 characters). Script will pad with spaces to maximum length as required. Default is ‘ ‘
	SduVendor	Per [1] 3.13.2.5.1 Specified as a string (0-16 characters). Script will pad with spaces to maximum length as required. Default is ‘1234567890123456’
	SystemDesignation	Per [1] 3.13.2.5.1 Specified as a string (0-16 characters). Script will pad with spaces to maximum length as required. Default is ‘1234567890123456’
	SduHwPartNumber	Per [1] 3.13.2.5.1 Specified as a string (0-16 characters). Script will pad with spaces to maximum length as required. Default is ‘1234567890123456’
	SduSwPartNumber	Per [1] 3.13.2.5.1 Specified as a string (0-16 characters). Script will pad with spaces to maximum length as required. Default is ‘1234567890123456’
	AntennaHwPartNumber	Per [1] 3.13.2.5.1 Specified as a string (0-16 characters). Script will pad with spaces to maximum length as required. Default is ‘1234567890123456’
	AntennaSwPartNumber	Per [1] 3.13.2.5.1 Specified as a string (0-16 characters). Script will pad with spaces to maximum length as required. Default is ‘1234567890123456’
	AircraftTailNumber	Per [1] 3.13.2.5.1 Specified as a string (0-7 characters). Script will pad with spaces to maximum length as required. Default is ‘1234567’
	AircraftType	Per [1] 3.13.2.5.1 Specified as a string (0-7 characters). Script will pad with spaces to maximum length as required. Default is ‘1234567’
	FlightId	Per [1] 3.13.2.5.1 Specified as a string (0-6 characters). Script will pad with spaces to maximum length as required. Default is ‘ ’

Section	Parameter	Values/Notes
	ServiceType	000: Aero-L 001: Aero-I 010: Aero-H 011: Aero-H+ 1xx Reserved for future use Will be based on the UT class (Class 4 = LGA, Class 6 = H+, Class 7 = IGA). Default is 3 (Aero-H+)
[Irs]	Latitude	-90 ≤ Latitude < +90 Positive = north Negative = south Default is 51 Note: Scripts may perturb by ± 0.703125 degrees.
	Longitude	-180 ≤ Longitude < +180 Positive = east Negative = west Default is -3 Note: Scripts may perturb by ± 0.703125 degrees.
	GroundSpeed	0 to 4096 Default is 100 Note: Scripts may perturb by ± 23 knots.
	TrueHeading	-180 ≤ TrueHeading < +180 Positive = east Negative = west Will be used for both Track and Heading labels. Default is 21 Note: Scripts may perturb by ± 1.40625 degrees.
	PitchAngle	-180 ≤ PitchAngle < +180 Positive for nose down (note 1). Default is 32
	RollAngle	-180 ≤ RollAngle < +180 Positive for left wing down (note 1). Default is 43
	Altitude	-131072 ≤ Altitude < 131072 Default is 5000 Note: Scripts may perturb by ± 32 feet.

Section	Parameter	Values/Notes
[Acars]	DefaultLength	In range of MinLength to MaxLength ACARS messages are sent as a modulo 256 counting sequence, starting with a random value. Default is 23
	MinLength	Defines the minimum length of an ACARS message. Default is 17
	MaxLength	Defines the maximum length of an ACARS message. Default is 238
[ConfigParams]	SdmAcT1	Per [1] 3.13.2.5.10 Default is 300
	ActualAcT1	Optional parameter indicating value to use for testing; defaults to SdmAcT1 if not present. This parameter is not defined by default.
	SdmAcT2	Per [1] 3.13.2.5.10 Default is 30
	ActualAcT2	Optional parameter indicating value to use for testing; defaults to SdmAcT2 if not present. This parameter is not defined by default.
	SdmAcT3	Per [1] 3.13.2.5.10 Default is 60
	ActualAcT3	Optional parameter indicating value to use for testing; defaults to SdmAcT3 if not present. This parameter is not defined by default.
	SdmAcT4	Per [1] 3.13.2.5.10 Default is 1200
	ActualAcT4	Optional parameter indicating value to use for testing; defaults to SdmAcT4 if not present. This parameter is not defined by default.
	SdmAcR1	Per [1] 3.13.2.5.10 Default is 1
	ActualAcR1	Optional parameter indicating value to use for testing; defaults to SdmAcR1 if not present. This parameter is not defined by default.
	SdmAcR2	Per [1] 3.13.2.5.10 Default is 2
	ActualAcR2	Optional parameter indicating value to use for testing; defaults to SdmAcR2 if not present. This parameter is not defined by default.

Section	Parameter	Values/Notes
	SdmAcR3	Per [1] 3.13.2.5.10 Default is 1
	ActualAcR3	Optional parameter indicating value to use for testing; defaults to SdmAcR3 if not present. This parameter is not defined by default.
	SdmAcR4	Per [1] 3.13.2.5.10 Reserved. Not used in phases 1-3. Default is 1
	ActualAcR4	Optional parameter indicating value to use for testing; defaults to SdmAcR4 if not present. Reserved. Not used in phases 1-3. This parameter is not defined by default.
	SdmAcR5	Per [1] 3.13.2.5.10 Default is 1
	ActualAcR5	Optional parameter indicating value to use for testing; defaults to SdmAcR5 if not present. This parameter is not defined by default.
	SdmAcF1	Per [1] 3.13.2.5.10 Default is $ff
	ActualAcF1	Optional parameter indicating value to use for testing; defaults to SdmAcF1 if not present. This parameter is not defined by default.
[Gateway]	AggwId	Per [1] 3.13.2.4.2 Default is 3
	DpId	Per [1] 3.13.2.4.2 Default is 4
	CspId	Per [1] 3.13.2.4.2 Default is 5
	GesId	Per [1] 3.13.2.4.2 Default is 6
[SBB002_110]	AlternateAcR5	Test-specific [ConfigParams]. Default is 5
	AlternateAcT2	Test-specific [ConfigParams]. Default is 3
	AlternateAcT3	Test-specific [ConfigParams]. Default is 5
[SBB002_112_1]	SatId	Test-specific [SDU] parameters used for verifying fields. Default is 1
	SpotId	Test-specific [SDU] parameters used for verifying fields. Default is 0

Section	Parameter	Values/Notes
	Imsi	Test-specific [SDU] parameters used for verifying fields. Default is '567890123'
	ImeiSvn	Test-specific [SDU] parameters used for verifying fields. Default is '2345678901'
	TerminalType	Test-specific [SDU] parameters used for verifying fields. Default is $01
	TypeApprovalId	Test-specific [SDU] parameters used for verifying fields. Default is ‘TA_ONE’
	SduVendor	Test-specific [SDU] parameters used for verifying fields. Default is ‘VENDOR_ONE’
	SystemDesignation	Test-specific [SDU] parameters used for verifying fields. Default is ‘SYSTEM_ONE’
	SduHwPartNumber	Test-specific [SDU] parameters used for verifying fields. Default is 'HARDWARE_ONE'
	SduSwPartNumber	Test-specific [SDU] parameters used for verifying fields. Default is 'SOFTWARE_ONE'
	AntennaHwPartNumber	Test-specific [SDU] parameters used for verifying fields. Default is 'ANT_HW_PN_ONE'
	AntennaSwPartNumber	Test-specific [SDU] parameters used for verifying fields. Default is 'ANT_SW_PN_ONE'
	AircraftTailNumber	Test-specific [SDU] parameters used for verifying fields. Default is 'TLN_ONE'
	AircraftType	Test-specific [SDU] parameters used for verifying fields. Default is 'TYPE_1'
	FlightId	Test-specific [SDU] parameters used for verifying fields. Default is 0

Section	Parameter	Values/Notes
[SBB002_112_2]	SatId	Test-specific [SDU] parameters used for verifying fields. Default is 7
	SpotId	Test-specific [SDU] parameters used for verifying fields. Default is 255
	Imsi	Test-specific [SDU] parameters used for verifying fields. Default is '123412345'
	ImeiSvn	Test-specific [SDU] parameters used for verifying fields. Default is '890089012'
	TerminalType	Test-specific [SDU] parameters used for verifying fields. Default is $8F
	TypeApprovalId	Test-specific [SDU] parameters used for verifying fields. Default is ‘TA_TWO’
	SduVendor	Test-specific [SDU] parameters used for verifying fields. Default is ‘VENDOR_TWO’
	SystemDesignation	Test-specific [SDU] parameters used for verifying fields. Default is ‘SYSTEM_TWO’
	SduHwPartNumber	Test-specific [SDU] parameters used for verifying fields. Default is 'HARDWARE_TWO'
	SduSwPartNumber	Test-specific [SDU] parameters used for verifying fields. Default is 'SOFTWARE_TWO'
	AntennaHwPartNumber	Test-specific [SDU] parameters used for verifying fields. Default is 'ANT_HW_PN_TWO'
	AntennaSwPartNumber	Test-specific [SDU] parameters used for verifying fields. Default is 'ANT_SW_PN_TWO'
	AircraftTailNumber	Test-specific [SDU] parameters used for verifying fields. Default is 'TLN_TWO'
	AircraftType	Test-specific [SDU] parameters used for verifying fields. Default is 'TYPE_2'

Section	Parameter	Values/Notes
	FlightId	Test-specific [SDU] parameters used for verifying fields. Default is '123456'
[SBB002_113]	Enable	TRUE to enable the test, FALSE to disable. This test verifies location formatting for the bounds of all parameters. Enable should be set to FALSE when the SDU will not be able to handle wide variations in location parameters; e.g., when in normal operational mode such as for Phase 3. Default TRUE.
[SBB003]	AlternateAcR2	Test-specific [ConfigParams]. Default is 5
	AlternateAcR3	Test-specific [ConfigParams]. Default is 3
	AlternateAcT2	Test-specific [ConfigParams]. Default is 5
[SBB005_102]	AlternateAcR3	Test-specific [ConfigParams]. Default is 4
	AlternateAcT2	Test-specific [ConfigParams]. Default is 10
[SBB012_104]	AlternateAcT1	Test-specific [ConfigParams]. Default is 3
	AlternateAcT2	Test-specific [ConfigParams]. Default is 7
	AlternateAcR3	Test-specific [ConfigParams]. Default is 1
[SBB012_105]	AlternateAcT1	Test-specific [ConfigParams]. Default is 10
	AlternateAcT2	Test-specific [ConfigParams]. Default is 10
	AlternateAcR3	Test-specific [ConfigParams]. Default is 4
[SBB012_106]	AlternateAcT1	Test-specific [ConfigParams]. Default is 12
	AlternateAcT2	Test-specific [ConfigParams]. Default is 12
	AlternateAcR3	Test-specific [ConfigParams]. Default is 1
	AlternateAcF1	Test-specific [ConfigParams]. Default is 0

Section	Parameter	Values/Notes
[SBB013_104]	AlternateAcR2	Test-specific [ConfigParams]. Default is 3
	AlternateAcT2	Test-specific [ConfigParams]. Default is 7
[ScriptTimers]	DefaultResponseTimeout	Default timeout for responses to ground-initiated AIGI or DNS messages; e.g., timeout for reception of ac_conf_ack after transmission of gw_conf from the Tester or timeout for reception of ac_logon_rq following DNS response from the Tester. Default 5 seconds. Could be adjusted for specific test configuration; e.g., Phase 1 will be quicker than Phase 2 or 3.
	DefaultControlResponse Timeout	Default timeout for responses to Control messages; e.g., timeout for reception of a DNS request following transmission of a Restart AAGW message from the Tester. Default 5 seconds. Could be much longer if [Control]TransportOverIp is False.
	DefaultAcarsEventTimeout	Default timeout for ACARS messaging transiting the system; e.g., AAGW should report an ACARS message from ground to the Williamsburg port within DefaultAcarsEventTimeout of gw_acars being sent from the Tester, or vice versa. Default 5 seconds. Could be adjusted for specific test configuration; e.g., Phase 1 will be quicker than Phase 2 or 3.
	DefaultStatusEventTimeout	Default timeout for a Status event to be reflected at the AIGI interface; e.g., AAGW should set NOCOMM to CMU within DefaultStatusEventTimeout of gw_msg_nak being sent from the Tester. Default 4 seconds. Could be adjusted for specific test configuration; e.g., Phase 1 will be quicker than Phase 2 or 3.
	TimerEarlyToleranceMs	Time in milliseconds by which a timer measurement can be less than its nominal and still be considered valid. Default 200 ms.
	TimerLateToleranceMs	Time in milliseconds by which a timer measurement can be greater than its nominal and still be considered valid. Must account for timing jitter, scheduling delays etc. in both the Tester and the transmission path. Default 1000 ms.

Section	Parameter	Values/Notes
[TestControl]	RestartAagwEachTest	True: Issue a Restart AAGW message at the start of each testcase. False: Only issue a Restart AAGW message at the start of those testcases that require it, or if AcT1 + (AcR1 * AcT2) has elapsed or if the SDU is indicating NOCOMM. Default False.
	StopSuiteOnFail	True: Test suite will stop when test fails. False: Test suite will not stop when test fails. Default False.
	SecsToWaitAfterTestFail	Time to wait (in seconds) at the end of a test if the test fails. Allows for transients to clear. Default 10.
	LogDir	Root directory for test case log files. Default “D:\Bplt_Log\SBB”.

[1] Sense was taken from text of US patent 7,487,014 although it is not important to the scripts since the specified value is simply passed on to the SDU via ARINC-429.

6. PHASE 1 OPERATION

6.1 Setup

An example Phase 1 test configuration is shown in Figure 6-1.

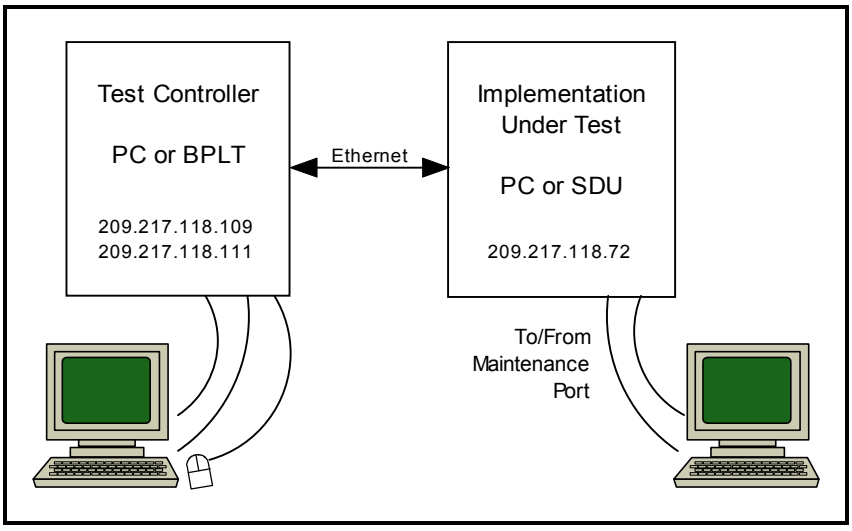


Figure 6-1 Example Phase 1 Test Configuration

Before running the test scripts for the first time, they must be correctly installed as described in Section 3.3.2 . The ini file parameters must be configured as required. This is done by creating D:\Bplt_dat\SBB\Ini\SBB_TestPar.ini with entries defined in Table 6-1.

Table 6-1 Typical Set of Parameters to be Defined in D:\Bplt_Dat\SBB\Ini\SBB_Par.ini for Phase 1

Section	Parameter	Values/Notes
[Control]	TransportOverIp	This should be set to TRUE
	IutControlIpAddress	IP address of the IUT; i.e., IP address to which to send control messages from the script. Set to 209.217.118.72 for the example shown in Figure 6-1.
[Arinc429]	TransportOverIp	This should be set to TRUE
	LocalIpAddress	IP address used for ARINC 429 interfaces. Set to 209.217.118.109 or 209.217.118.111 for the example shown in Figure 6-1.
	IutArincIpAddress	IP address of the IUT; i.e., IP address to which to send ARINC 429 messages from the script. Set to 209.217.118.72 for the example shown in Figure 6-1.

Section	Parameter	Values/Notes
[Aigi]	PrimaryAggwIpAddress	Default IP address of the AGGW (Tester). Set to 209.217.118.109 for the example shown in Figure 6-1.
	AlternateAggwIpAddress	Alternate IP address of the AGGW (Tester). Set to 209.217.118.111 for the example shown in Figure 6-1.

The Test Controller software should be running, and the IUT must be configured correctly and connected, ready for the execution of the tests.

6.2 Script and Suite Files

The SBB Test Suites exercise the AIGI interface between AAGW and AGGW. These tests are designed to run in an automated manner with the minimum of user intervention.

The test script and suite files are contained in the directory `C:\Bplt_Dat\SBB`. Within this directory are subdirectories for each section (`\SBB001`, `\SBB002`, … `SBB022`). Each subdirectory contains individual SBB script files for each testcase, e.g. `SBB001_101.scr` as well as any header files used by the script files in the subdirectory e.g. `SBB001.inc`.

For automated execution of all the SBB tests, a suite file referencing all of the scripts in each subdirectory is provided in `C:\Bplt_Dat\SBB\Suite\RunAll.sui`.

6.3 Initialization and AES Logon Status

When any of the SBB tests is run for the first time, it assumes that the AAGW under test is not yet logged on. Prior to first execution, the AAGW should be reset so that it also believes it is not yet logged on. In Phase 1, when any SBB test is run with no AAGW logged on, the Tester will send a RESTART message to the AAGW to trigger AAGW to search for the AGGW via DNS and then logon to the AG GW when it receives the DNS response. When the AAGW has been successfully logged on, the Tester will timestamp the AAGW logon event and execute the test. If the AAGW should become inactive for greater than the configurable inactivity timeout, the Tester will assume that the AAGW is no longer logged on: the next script being run will also cause a RESTART message to be sent to the AAGW.

If a situation occurs in which the AAGW believes it is not logged on, the SBB test suite may need to be reset to view the AAGW as logged off. This can be accomplished by running the script `C:\Bplt_Dat\SBB\Reset.scx`. The AAGW will then receive a RESTART message at the start of the next test.

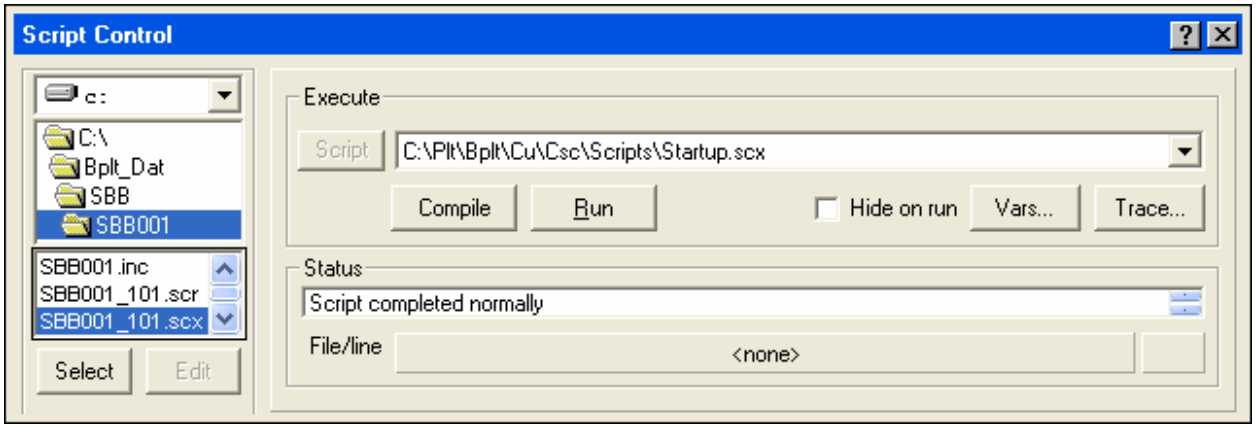
Should the AAGW system configuration (e.g. IP address or port number) be required to change in the course of execution of any SBB test or any SBB suite, the Tester should be reset to ensure that both AAGW and Tester use the new configuration.

6.4 Test Script Execution

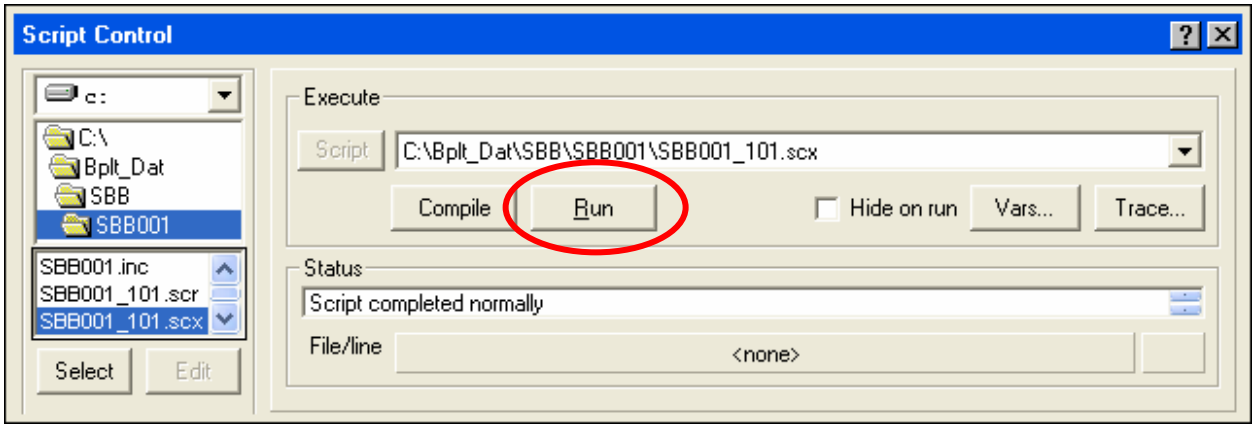
The steps below indicate how a test script is run:

1) In the BPLT window, click on the Script button.

2) In the Script Control window, navigate to the subdirectory containing the script that is to be run.

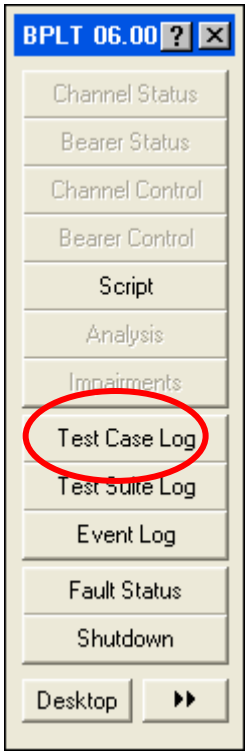


3) Double-click the script file to be executed (or highlight it and then click Select). The script name should appear in the Execute frame. Then click the Run button.

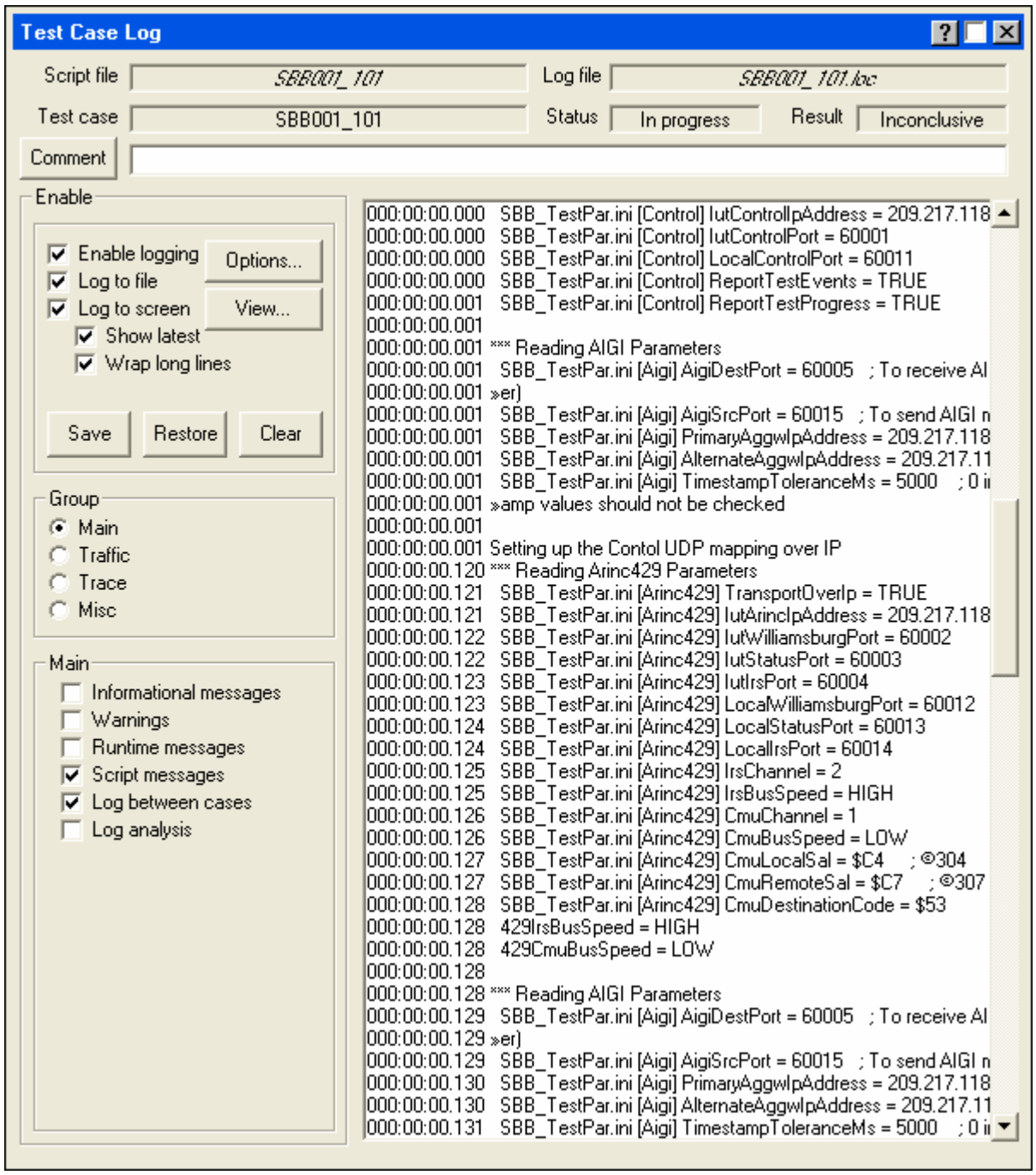


Note: Files with extension .scr and script file source code and files with extension .scx are compiled scripts. Either can be executed; however, pre-compiled scripts will start running much more quickly. It is recommended that all scripts be pre-compiled as outlined in Table 3-2.

4) The test case logs can be displayed by clicking the “Test Case Log” button on the BPLT menu window.



This causes the Test Case Log window to pop-up:


Test Case Log
Script file
SBB001_101
Log file
SBB001_101.loc
Test case
SBB001_101
Status
In progress
Result
Inconclusive
Comment
Enable
Enable logging
Log to file
Log to screen
Show latest
Wrap long lines
Options...
View...
Save
Restore
Clear
Group
Main
Traffic
Trace
Misc
Main
Informational messages
Warnings
Runtime messages
Script messages
Log between cases
Log analysis
000:00:00.000 SBB_TestPar.ini [Control] IutControlIpAddress = 209.217.118
000:00:00.000 SBB_TestPar.ini [Control] IutControlPort = 60001
000:00:00.000 SBB_TestPar.ini [Control] LocalControlPort = 60011
000:00:00.000 SBB_TestPar.ini [Control] ReportTestEvents = TRUE
000:00:00.001 SBB_TestPar.ini [Control] ReportTestProgress = TRUE
000:00:00.001
000:00:00.001 *** Reading AIGI Parameters
000:00:00.001 SBB_TestPar.ini [Aigi] AigiDestPort = 60005 ; To receive AI
000:00:00.001 »er)
000:00:00.001 SBB_TestPar.ini [Aigi] AigiSrcPort = 60015 ; To send AIGI n
000:00:00.001 SBB_TestPar.ini [Aigi] PrimaryAggwIpAddress = 209.217.118
000:00:00.001 SBB_TestPar.ini [Aigi] AlternateAggwIpAddress = 209.217.11
000:00:00.001 SBB_TestPar.ini [Aigi] TimestampToleranceMs = 5000 ; 0 i
000:00:00.001 »amp values should not be checked
000:00:00.001
000:00:00.001 Setting up the Contol UDP mapping over IP
000:00:00.120 *** Reading Arinc429 Parameters
000:00:00.121 SBB_TestPar.ini [Arinc429] TransportOverIp = TRUE
000:00:00.121 SBB_TestPar.ini [Arinc429] IutArincIpAddress = 209.217.118
000:00:00.122 SBB_TestPar.ini [Arinc429] IutWilliamsburgPort = 60002
000:00:00.122 SBB_TestPar.ini [Arinc429] IutStatusPort = 60003
000:00:00.123 SBB_TestPar.ini [Arinc429] IutIrsPort = 60004
000:00:00.123 SBB_TestPar.ini [Arinc429] LocalWilliamsburgPort = 60012
000:00:00.124 SBB_TestPar.ini [Arinc429] LocalStatusPort = 60013
000:00:00.124 SBB_TestPar.ini [Arinc429] LocalIrsPort = 60014
000:00:00.125 SBB_TestPar.ini [Arinc429] IrsChannel = 2
000:00:00.125 SBB_TestPar.ini [Arinc429] IrsBusSpeed = HIGH
000:00:00.126 SBB_TestPar.ini [Arinc429] CmuChannel = 1
000:00:00.126 SBB_TestPar.ini [Arinc429] CmuBusSpeed = LOW
000:00:00.127 SBB_TestPar.ini [Arinc429] CmuLocalSal = $C4 ; ©304
000:00:00.127 SBB_TestPar.ini [Arinc429] CmuRemoteSal = $C7 ; ©307
000:00:00.128 SBB_TestPar.ini [Arinc429] CmuDestinationCode = $53
000:00:00.128 429IrsBusSpeed = HIGH
000:00:00.128 429CmuBusSpeed = LOW
000:00:00.128
000:00:00.128 *** Reading AIGI Parameters
000:00:00.129 SBB_TestPar.ini [Aigi] AigiDestPort = 60005 ; To receive AI
000:00:00.129 »er)
000:00:00.129 SBB_TestPar.ini [Aigi] AigiSrcPort = 60015 ; To send AIGI n
000:00:00.130 SBB_TestPar.ini [Aigi] PrimaryAggwIpAddress = 209.217.118
000:00:00.130 SBB_TestPar.ini [Aigi] AlternateAggwIpAddress = 209.217.11
000:00:00.131 SBB_TestPar.ini [Aigi] TimestampToleranceMs = 5000 ; 0 i

6.5 Test Script Output

The execution of any SBB .scr or .scx file causes a Test Case Log file (.loc file) to be generated with the name of the test file, e.g., running SBB001_101.scx results in SBB001_101.loc in the D:\Bplt_Log\SBB directory. System event logs, which may be useful for debugging, are named Event*nnn*.log (e.g. Event022.log) and can be found in D:\Bplt_Log. See the BPLT Operator's Manual [14] for more information.

6.6 Fault Isolation

If a test fails, the first course of action should be to examine the test case logs in the test case log directory. The failure cause is usually explained near the bottom of the log file.

6.6.1 Connectivity Issues

For basic connectivity issues, the key is ensuring that the networking has been set up properly and that all of the IP addresses are consistent between the Tester, IUT and the Tester ini file.

If a mandatory test case parameter with no default, such as an IP address, is not specified, a run-time error message box such as shown in Figure 6-2 will be displayed.

Figure 6-2 Example Run-Time Error Message For Bad IP Address

All IP/ports in the INI file must be non-overlapping so that the UDP ports can be created. If this is not the case, a run-time error message box such as shown in Figure 6-3 will be displayed.

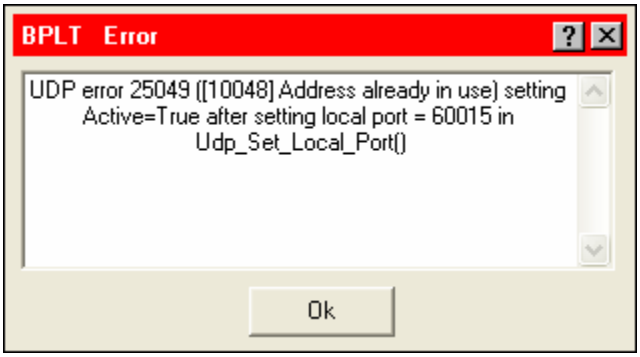

BPLT Error
UDP error 25049 ([10048] Address already in use) setting Active=True after setting local port = 60015 in Udp_Set_Local_Port()
Ok

Figure 6-3 Example Run-Time Error Message For Overlapping Ports

The error shown in Figure 6-3 can also be displayed if the port numbers are being changed. If this occurs, the PC can be rebooted. Alternatively, one can reassign the affected ports to port numbers that have not yet been used, rerun the script, and when it fails restore the affected ports to the desired values.

If the IP addresses are correct, but no UDP packets are getting through, verify that any firewalls that are present are correctly configured to pass the specified port numbers.

6.7 Test Suite Execution

For automated execution of all the SBB tests, a suite file referencing all of the scripts in each subdirectory is provided in `C:\Bplt_Dat\SBB\Suite\RunAll.sui`.

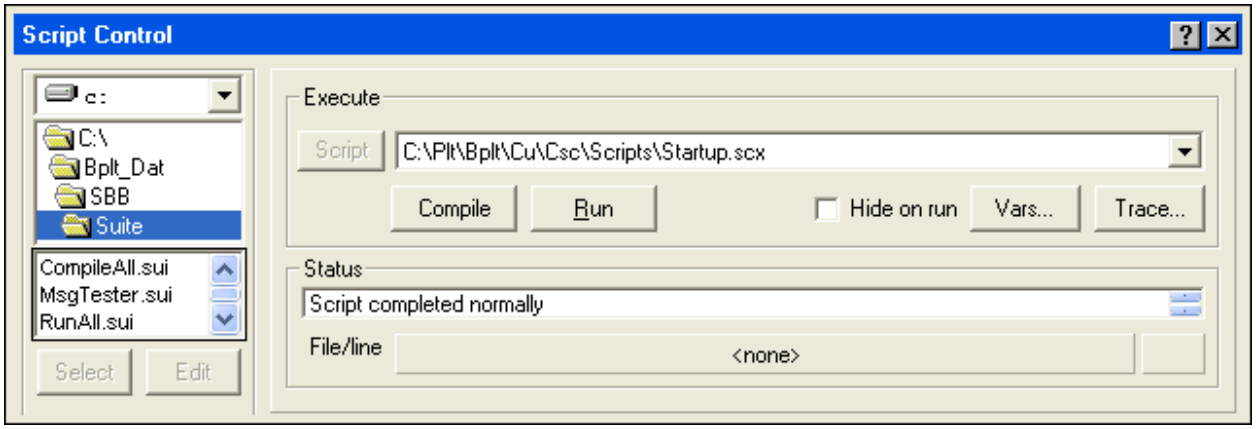
If the AAGW is to receive a RESTART command at the start of each test (i.e., and go through the process of locating AGGW and logging on to AGGW), the ini file should be edited so that RestartAagwEachTest (in [TestControl]) is TRUE.

If the AAGW is only meant to receive a RESTART command from each test script as needed (i.e. when logged out), then this parameter should be set to FALSE. This is the default setting.

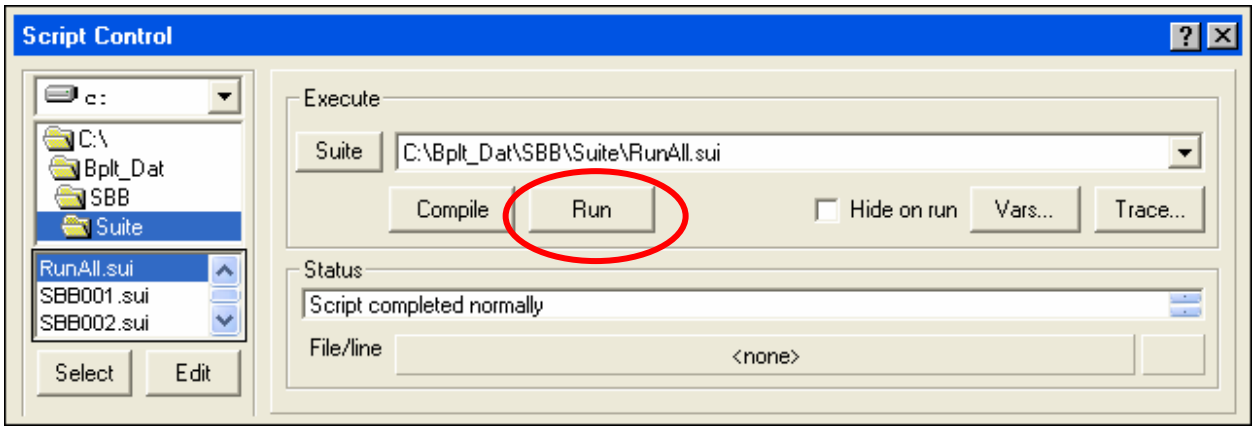
In order to exercise the AIGI interface messages that do not include position reporting (i.e., the _n messages), the ini file should have the ReportLocation parameter (in [Sdu]) , set to FALSE. In general, it is recommended that the entire suite be run twice – once with [Sdu]ReportLocation set to TRUE (the default) and once with it set to FALSE.

The following steps should be used to execute the full test suite.

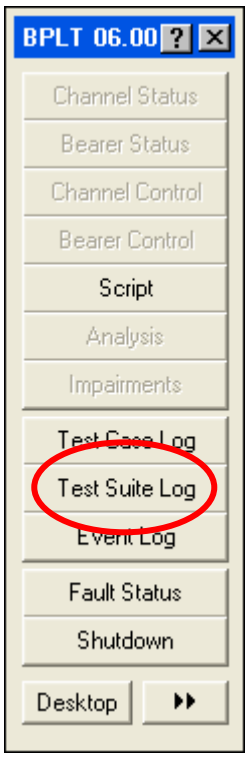
1) In the Script Control window, navigate to the subdirectory containing the suite that is to be run.



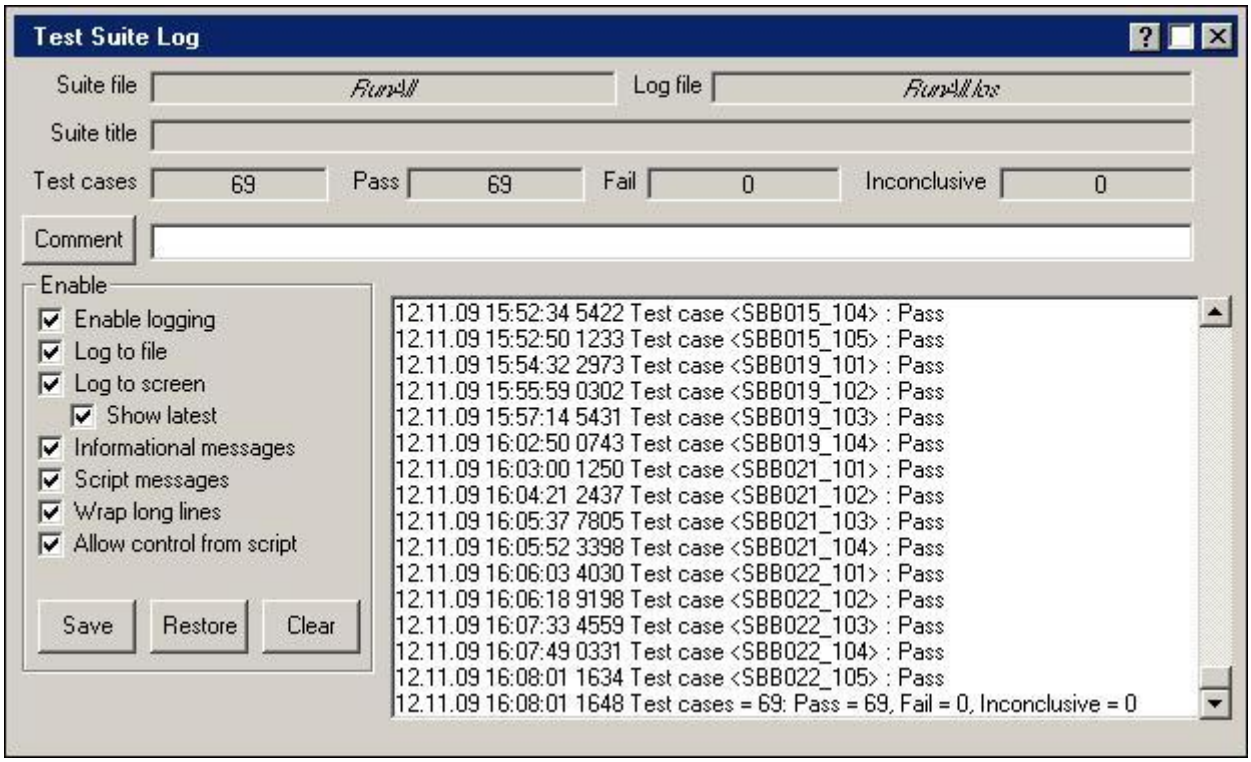
2) Double-click the suite file to be executed (or highlight it and then click Select). The script name should appear in the Execute frame. Then click the Run button.



3) The progress of a test suite can be displayed by clicking the “Test Suite Log” button on the BPLT menu window.



This causes the Test Suite Log window to pop-up:



7. PHASE 2 OPERATION

In Phase 2, the CMU and IRS interfaces are implemented via ARINC-429 rather than via IP. The scripts themselves are identical to those used in Phase 1. Only the hardware configuration, and some associated ini file parameters need to change. In general, except as noted in the subsections below, the instructions in section 6 apply.

7.1 Setup

An example Phase 2 test configuration is shown in Figure 7-1.

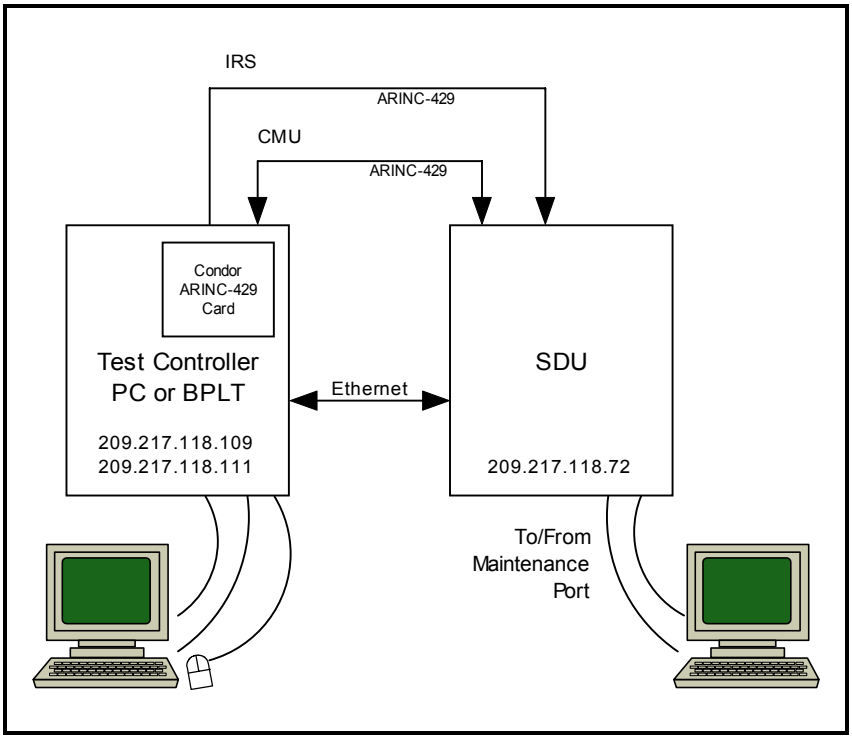


Figure 7-1 Example Phase 2 Test Configuration

As with Phase 1, before running the test scripts for the first time, they must be correctly installed as described in Section 3.3.2 . The ini file parameters must be configured as required. This is done by creating D:\Bplt_dat\SBB\Ini\SBB_TestPar.ini with entries defined in Table 7-1.

Table 7-1 Typical Set of Parameters to be Defined in D:\Bplt_Dat\SBB\Ini\SBB_Par.ini for Phase 2

Section	Parameter	Values/Notes
[Control]	TransportOverIp	This should be set to TRUE
	IutControlIpAddress	IP address of the IUT; i.e., IP address to which to send control messages from the script. Set to 209.217.118.72 for the example shown in Figure 7-1.
[Arinc429]	TransportOverIp	This should be set to FALSE
	IrsChannel	ARINC 429 logical transmit channel for IRS labels transmitted from the Tester. Set as per wired connections. Default is 2.
	IrsBusSpeed	Set to 'HIGH' or 'LOW' as appropriate. Default is 'HIGH'.
	CmuChannel	ARINC 429 logical transmit channel for CMU label 270 and Williamsburg transmitted from the tester. ARINC 429 logical receive channel for SDU label 270 and Williamsburg received by the Tester. Set as per wired connections. Default is 1.
	CmuBusSpeed	Set to 'HIGH' or 'LOW' as appropriate. Default is 'LOW'.
[Aigi]	PrimaryAggwIpAddress	Default IP address of the AGGW (Tester). Set to 209.217.118.109 for the example shown in Figure 7-1.
	AlternateAggwIpAddress	Alternate IP address of the AGGW (Tester). Set to 209.217.118.111 for the example shown in Figure 7-1.

The Tester must be connected to the IUT via ARINC 429. Two channels are used – one for the CMU interface and one for the IRS interface. See section 3.2.1 for installation and cabling information.

By default, the Tester assumes that the first two logical ARINC 429 channels correspond to the first two physical ARINC 429 channels on the CEI-520A. If this is not the case, the logical to physical channel mapping in the [Channels] section of Car_ex.ini must be modified.

7.2 Test Execution

Test script and test suite execution is as described in sections 6.2 through 6.5 above.

7.3 Fault Isolation

As in Phase 1, if a test fails, the first course of action should be to examine the test case logs in the test case log directory. The failure cause is usually explained near the bottom of the log file.

For failures involving the CMU interface, D:\Bplt_log\Wbp_err*n*.log can also be examined, where *n* is the ARINC logical channel number. More detailed logging of ARINC labels on the CMU interface can be obtained by creating the following entry in D:\Bplt_dat\Ini\Car_Ex.ini:

```
[Wbp]
EnableCommLogging=True
```

The resulting log file can be found in D:\Bplt_log\Wbp_comm*n*.log, where *n* is the ARINC logical channel number. By default, both Wbp_err.log and Wbp_comm.log are deleted when the Tester software is restarted. This can be overridden by creating the following entry in D:\Bplt_dat\Ini\Car_Ex.ini:

```
[Wbp]
DeleteLogsAtStart=False
```

If basic communication is not established, double-check that the ARINC signals are correctly wired and that the appropriate ARINC speed has been selected.

8. PHASE 3 OPERATION

In Phase 3, the CMU and IRS interfaces are implemented via ARINC-429 rather than via IP (as in Phase 2) and the AIGI interface is implemented over the SBB air interface rather than via IP. The scripts themselves are identical to those used in Phases 1 and 2. Only the hardware configuration, and some associated ini file parameters need to change. In general, except as noted in the subsections below, the instructions in section 6 apply.

8.1 Setup

An example Phase 3 test configuration, assuming the use of a Gatehouse BGAN Network Emulator, is shown in Figure 8-1. Other configurations that don't utilize the BNE are also possible.

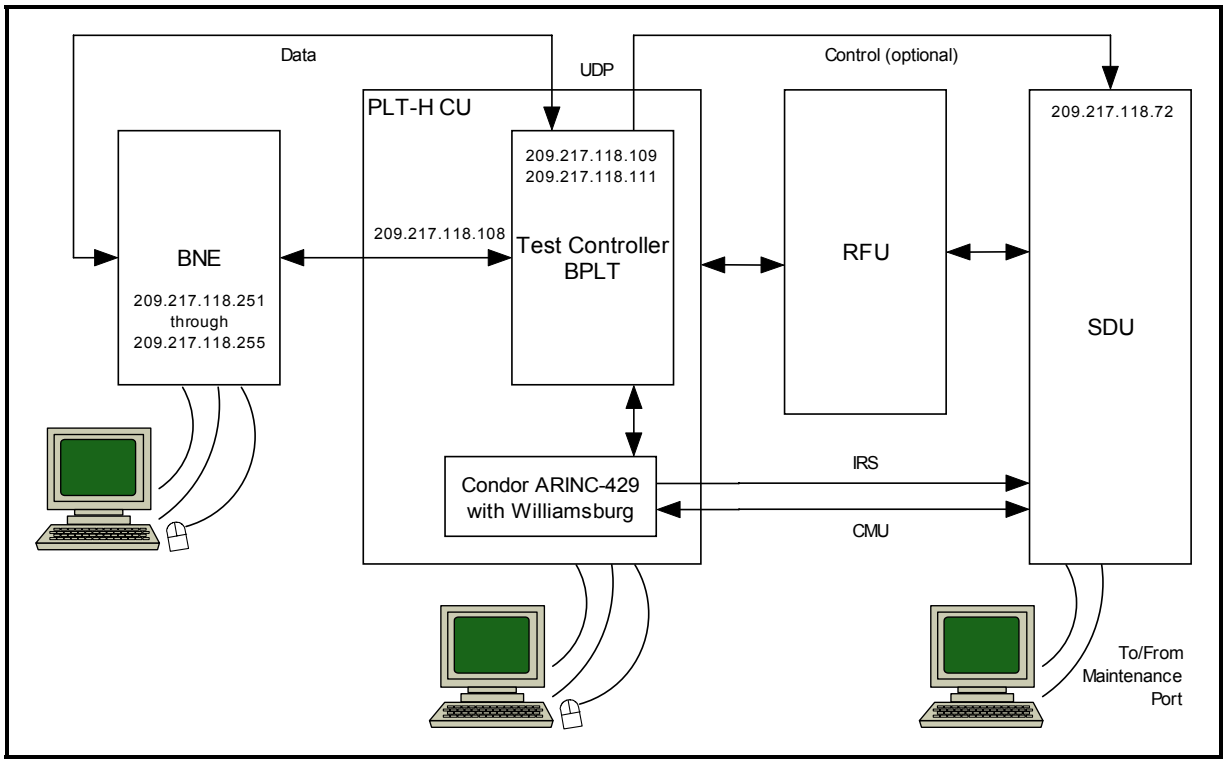


Figure 8-1 Example Phase 3 Test Configuration

In the example configuration shown, the BNE is configured to use IP addresses 209.218.118.251 through 209.217.118.255 for itself, the BPLT is configured to use 209.217.118.108 as its protocol tester (BNE) IP address and 209.217.118.109 and 209.217.118.111 as its primary and secondary AGGW/DNS addresses respectively. Note that the BNE cannot run on the BPLT platform due to the mechanisms that the BNE employs for sending and receiving IP packets.

As with Phase 1, before running the test scripts for the first time, they must be correctly installed on the BPLT as described in Section 3.3.2 . The ini file parameters must be configured as required. This is done by creating D:\Bplt_dat\SBB\Ini\SBB_TestPar.ini with entries defined in Table 8-1.

Table 8-1 Typical Set of Parameters to be Defined in D:\Bplt_Dat\SBB\Ini\SBB_Par.ini for Phase 3

Section	Parameter	Values/Notes
[Control]	TransportOverIp	If the SDU implements the Control interface, this should be set to TRUE; otherwise, it should be set to FALSE in which case the scripts will prompt the user to configure the SDU appropriately. Set to 209.217.118.72 for the example shown in Figure 8-1.
	IutControlIpAddress	IP address of the IUT; i.e., IP address to which to send control messages from the script.
[Arinc429]	TransportOverIp	This should be set to FALSE
	IrsChannel	ARINC 429 logical transmit channel for IRS labels transmitted from the Tester. Set as per wired connections. Default is 2.
	IrsBusSpeed	Set to 'HIGH' or 'LOW' as appropriate. Default is 'HIGH'.
	CmuChannel	ARINC 429 logical transmit channel for CMU label 270 and Williamsburg transmitted from the tester. ARINC 429 logical receive channel for SDU label 270 and Williamsburg received by the Tester. Set as per wired connections. Default is 1.
	CmuBusSpeed	Set to 'HIGH' or 'LOW' as appropriate. Default is 'LOW'.
[Aigi]	PrimaryAggwIpAddress	Default IP address of the AGGW (Tester). Set to 209.217.118.109 for the example shown in Figure 8-1.
	AlternateAggwIpAddress	Alternate IP address of the AGGW (Tester). Set to 209.217.118.111 for the example shown in Figure 8-1.
[Dns]	PrimaryDnsIpAddress	Set to the IP address of the primary DNS server that will be provided for PDP contexts that are set up by the AAGW. This must be an IP address on the Tester. Set to 209.217.118.111 for the example shown in Figure 8-1.
	SecondaryDnsIpAddress	Set to the IP address of the secondary DNS server that will be provided for PDP contexts that are set up by the AAGW. This must be an IP address on the Tester. Set to 209.217.118.108 for the example shown in Figure 8-1.
[Sdu]	IcaoAddress	If the SDU does not implement the Control interface, set to match the SDU configuration. If the Control interface is implemented, setting these parameters is optional since the scripts will communicate the
	ProtocolVersion	
	Imsi	

Section	Parameter	Values/Notes
	ImeiSvn	defaults to the SDU when restarting the AAGW.
	TerminalType	
	TypeApprovalId	
	SduVendor	
	SystemDesignation	
	SduHwPartNumber	
	SduSwPartNumber	
	AntennaHwPartNumber	
	AntennaSwPartNumber	
	AircraftTailNumber	
	AircraftType	
	FlightId	
	ServiceType	
	SatId	Set to the satellite ID used on the air interface.
	SpotId	Set to the spot beam ID used on the air interface. Should be consistent with the specified IRS position.
[Irs]	Latitude	Should be consistent with the desired satellite ID and spot beam ID. Note that the scripts may perturb the latitude and longitude by up to ± 0.703125 degrees, so care should be taken that this would not cause a change in spot beam.
	Longitude	
[SBB002_112_1]	SatId	For Phase 3, should be set to the same as [Sdu]SatId, so that it remains consistent with the location of the SDU.
	SpotId	For Phase 3, should be set to the same as [Sdu]SpotId, so that it remains consistent with the location of the SDU.
	Imsi	If the SDU does not implement the Control interface, set to match an alternate SDU configuration to the baseline one. If the Control interface is implemented, setting these parameters is optional since the scripts will communicate the default values to the SDU when restarting the AAGW.
	ImeiSvn	
	TerminalType	
	TypeApprovalId	
	SduVendor	
	SystemDesignation	
	SduHwPartNumber	
	SduSwPartNumber	
	AntennaHwPartNumber	
	AntennaSwPartNumber	
	AircraftTailNumber	

Section	Parameter	Values/Notes
	AircraftType	
	FlightId	
[SBB002_112_2]	SatId	For Phase 3, should be set to the same as [Sdu]SatId, so that it remains consistent with the location of the SDU.
	SpotId	For Phase 3, should be set to the same as [Sdu]SpotId, so that it remains consistent with the location of the SDU.
	Imsi	If the SDU does not implement the Control interface, set to match an alternate SDU configuration to the baseline one. If the Control interface is implemented, setting these parameters is optional since the scripts will communicate the default values to the SDU when restarting the AAGW.
	ImeiSvn	
	TerminalType	
	TypeApprovalId	
	SduVendor	
	SystemDesignation	
	SduHwPartNumber	
	SduSwPartNumber	
	AntennaHwPartNumber	
	AntennaSwPartNumber	
	AircraftTailNumber	
	AircraftType	
	FlightId	
[SBB002_113]	Enable	Set to FALSE to disable testing full ranges of IRS values. This avoids “moving” the SDU to invalid or unexpected positions that could cause beam handovers etc.
[ScriptTimers]	DefaultResponseTimeout	This should be at least 6 in Phase 3.
	DefaultControlResponse Timeout	This should be at least 6 in Phase 3.
	DefaultAcarsEventTimeout	This should be at least 7 in Phase 3.
	DefaultStatusEventTimeout	This should be at least 6 in Phase 3.
	TimerEarlyToleranceMs	This should be at least 2000 in Phase 3.
	TimerLateToleranceMs	This should be at least 2000 in Phase 3.

See section 7.1 for details on the ARINC 429 connections between the Tester and the IUT.

8.2 Test Execution

If testing with a nailed up connection, the SDU must have registered with the [emulated] network and established a PDP context. Once this is complete, scripts can be run as described in sections 6.2 through 6.5 above.

With a more complete AAGW implementation where the PDP context is not nailed up, the scripts will initiate an AAGW restart as required, which will cause the SDU/AAGW to register with the [emulated] network, establish a PDP context etc.

If the script times out waiting for the first DNS query following an AAGW restart, the parameter [ScriptTimers]DefaultControlResponseTimeout may need to be increased.

Also, if the IP control interface is not used, the user should manually initiate and acknowledge every restart, when needed. In order to minimize user interaction, it is recommended that [TestControl]RestartAagwEachTest be left at its default setting of FALSE.

8.3 Fault Isolation

As in Phases 1 and 2, if a test fails, the first course of action should be to examine the test case logs in the test case log directory. The failure cause is usually explained near the bottom of the log file.

Note that some tweaking of timer-related parameters may be required in order to compensate for link delays now that the scripts are communicating over the air interface.

If script fails up front due to a basic communication error between AAGW and the script (simulated AGGW), ensure that the SDU is able to register with the network and establish the PDP context. If this connection has been established, examine the networking configuration to ensure that IP packets will be routed appropriately.

If the script times out waiting for the first DNS query following an AAGW restart, the parameter [ScriptTimers]DefaultControlResponseTimeout may need to be increased. Alternatively, there may be a connectivity problem through the emulated network. Ensure that the UT is still acquired and that the PDP context is still active.